

本书第1版荣获“北京高等教育精品教材”

普通高等教育“十一五”国家级规划教材

丛书主编 谭浩强

高等院校计算机应用技术规划教材

实用技术教材系列

网站编程技术实用教程

（第2版）

邵丽萍 张后扬 郭春芳 编著

根据“中国高等院校计算机基础教育课程体系”组织编写



清华大学出版社

高等院校计算机应用技术规划教材

网站编程技术实用教程

（第 2 版）

邵丽萍　张后扬　郭春芳　编著

清华大学出版社

北　京

内 容 简 介

网站编程技术在 Internet 上有着广泛的应用。本书系统介绍了几种常用的网站编程技术：HTML、JavaScript、CSS、JSP、XML、MVC 等，主要内容包括 Web 开发技术概述，HTML 语言基本结构，JavaScript 的使用，CSS 样式表，使用 JSP 创建动态网页，数据库的使用，网站开发方法，使用模块化方法制作主页，网站应用程序实例等。

本书每个应用程序都有完整的程序源代码和在浏览器中程序运行显示的结果图，方便读者练习参考使用。

本书以通俗易懂的语言从实用的角度介绍了网站开发的几种编程技巧。不仅适合没有网站编程经验的读者学习，也适合有一定网站编程基础的读者学习。本书可作为高等院校、高职高专和网站编程技术培训班的使用教材，还可以作为网站编程爱好者制作网站的参考书。

图书在版编目（CIP）数据

网站编程技术实用教程/邵丽萍，张后扬，郭春芳编著. —2 版. —北京：清华大学出版社，2009.5
（高等院校计算机应用技术规划教材）
ISBN 978-7-302-19720-1

Ⅰ. 网…　Ⅱ. ①邵…②张…③郭…　Ⅲ. 主页制作—程序设计—高等学校—教材
Ⅳ. TP393.092

中国版本图书馆 CIP 数据核字(2009)第 037199 号

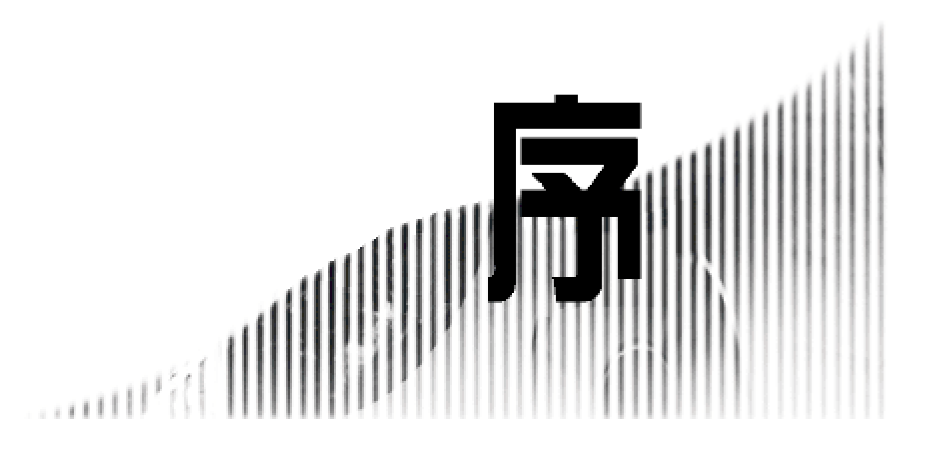
责任编辑：谢　琛　顾　冰
责任校对：时翠兰
责任印制：李红英

出版发行：清华大学出版社　　**地　　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：**100084
社　总　机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印　刷　者：北京市昌平环球印刷厂
装　订　者：三河市李旗庄少明装订厂
经　　销：全国新华书店
开　　本：185×260　　**印　张：**21.75　　**字　数：**494 千字
版　　次：2009 年 5 月第 2 版　　**印　次：**2009 年 5 月第 1 次印刷
印　　数：1～4000
定　　价：29.50 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：032179-01

编辑委员会

《高等院校计算机应用技术规划教材》



《高等院校计算机应用技术规划教材》

进入 21 世纪,计算机成为人类常用的现代工具,每一个有文化的人都应当了解计算机,学会使用计算机来处理各种的事务。

学习计算机知识有两种不同的方法:一种是侧重理论知识的学习,从原理入手,注重理论和概念;另一种是侧重于应用的学习,从实际入手,注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对多数人来说,计算机是作为一种工具来使用的,应当以应用为目的、以应用为出发点。对于应用性人才来说,显然应当采用后一种学习方法,根据当前和今后的需要,选择学习的内容,围绕应用进行学习。

学习计算机应用知识,并不排斥学习必要的基础理论知识,要处理好这二者的关系。在学习过程中,有两种不同的学习模式:一种是金字塔模型,亦称为建筑模型,强调基础宽厚,先系统学习理论知识,打好基础以后再联系实际应用;另一种是生物模型,植物并不是先长好树根再长树干,长好树干才长树冠,而是树根、树干和树冠同步生长的。对计算机应用性人才教育来说,应该采用生物模型,随着应用的发展,不断学习和扩展有关的理论知识,而不是孤立地、无目的地学习理论知识。

传统的理论课程采用以下的三部曲:提出概念—解释概念—举例说明,这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者,我们提倡新的三部曲:提出问题—解决问题—归纳分析。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

检查学习好坏的标准,不是“知道不知道”,而是“会用不会用”,学习的目的主要在于应用。因此希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力,启发学生的求知欲望。

全国高等院校计算机基础教育研究会历来倡导计算机基础教育必须坚持面向应用的正确方向，要求构建以应用为中心的课程体系，大力推广新的教学三部曲，这是十分重要的指导思想，这些思想在《中国高等院校计算机基础课程》中作了充分的说明。本丛书完全符合并积极贯彻全国高等院校计算机基础教育研究会的指导思想，按照《中国高等院校计算机基础教育课程体系》组织编写。

这套《高等院校计算机应用技术规划教材》是根据广大应用型本科和高职高专院校的迫切需要而精心组织的，其中包括 4 个系列：

(1) 基础教材系列。该系列主要涵盖了计算机公共基础课程的教材。

(2) 应用型教材系列。适合作为培养应用性人才的本科院校和基础较好、要求较高的高职高专学校的主干教材。

(3) 实用技术教材系列。针对应用型院校和高职高专院校所需掌握的技能技术编写的教材。

(4) 实训教材系列。应用型本科院校和高职高专院校都可以选用这类实训教材。其特点是侧重实践环节，通过实践(而不是通过理论讲授)去获取知识，掌握应用。这是教学改革的一个重要方面。

本套教材是从 1999 年开始出版的，根据教学的需要和读者的意见，几年来多次修改完善，选题不断扩展，内容日益丰富，先后出版了 60 多种教材和参考书，范围包括计算机专业和非计算机专业的教材和参考书；必修课教材、选修课教材和自学参考的教材。不同专业可以从中选择所需要的部分。

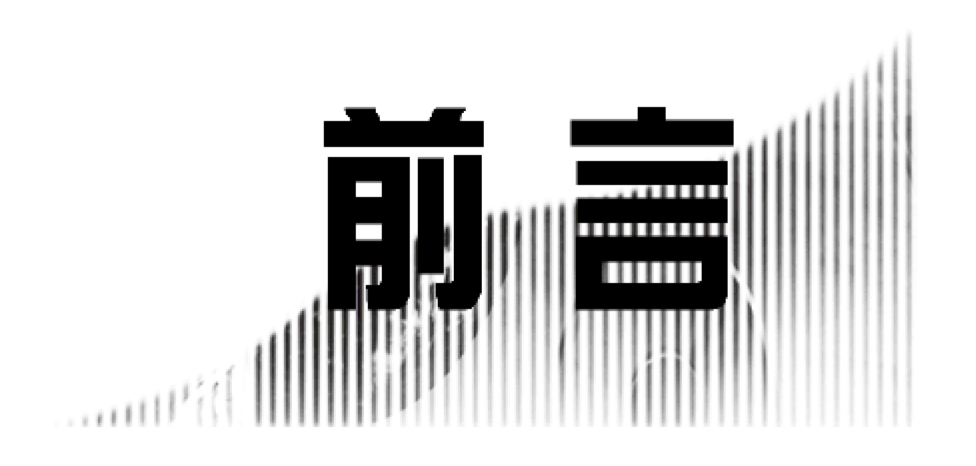
为了保证教材的质量，我们遴选了有丰富教学经验的高校优秀教师分别作为本丛书各教材的作者，这些老师长期从事计算机的教学工作，对应用型的教学特点有较多的研究和实践经验。由于指导思想明确、作者水平较高，教材针对性强，质量较高，本丛书问世 7 年来，愈来愈得到各校师生的欢迎和好评，至今已发行了 240 多万册，是国内应用型高校的主流教材之一。2006 年被教育部评为普通高等教育“十一五”国家级规划教材，向全国推荐。

由于我国的计算机应用技术教育正在蓬勃发展，许多问题有待深入讨论，新的经验也会层出不穷，我们会根据需要不断丰富本丛书的内容，扩充丛书的选题，以满足各校教学的需要。

本丛书肯定会有不足之处，请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长
《高等院校计算机应用技术规划教材》主编　**谭浩强**

2008 年 5 月 1 日于北京清华园



随着Internet的普及，网站编程技术越来越受到人们的重视，越来越多的企业和个人都希望使用网站编程技术，建立企业的网站和个人网页。为了迎接信息时代的挑战，学习和掌握网站编程技术无疑会带来更多的机遇和更大的发展空间。

本书第1版受到读者的欢迎，在2006年被评为北京市高等院校精品教材。由于Web开发技术在不断发展，为了体现Web开发技术的新特点，答谢读者的喜爱，满足读者的需要，作者对本书重新进行了修订。作为一本教材，本书第2版对网站编程的基本技术做了详细介绍，包括如何使用HTML语言编写网页，如何用JavaScript编写网页，如何给网页添加处理数据的功能，如何与用户进行交互操作，如何使用CSS样式表来设计页面的显示方式，还介绍了什么是动态网页(Active Server Page)，以及如何使用Access数据库技术和JSP技术编写动态网页。在内容的编排上体现了新的计算机教学思想和方法，以“提出问题→解决问题的方法和实例→归纳必要的结论和概念”的方式介绍了网站编程的基本思路。

1. 本书主要特色

1）通俗易懂、图文并茂

本书都是通过具体的例子来介绍有关Web开发技术的概念、方法和技术，每章都有大量完整的例子与源程序代码，用来说明使用网站编程的基本步骤和基本方法，并有图片配合说明，通俗易懂，读者完全可以按书中介绍的方法完成每个例子，通过实例理解网站编程技术的基本思想和编程技巧。

2）内容全面、结构清晰、循序渐进

本书对整个内容做了精心设计和安排，首先介绍HTML语言的产生、特点、结构，然后介绍脚本语言JavaScript、CSS、JSP、XML、MVC，接着介绍网站开发的阶段与开发方法，最后介绍一个具体的网站开发实现过程。循序渐进，先易后难，逐步深入，通过具体实例引出后续内容或巩固前面介绍的内容。

3）实践性强

在使用本书学习时，可结合具体的实例，上机实践，按照书中介绍的例子，在短时间内使用Web开发技术进行一个动态网站的开发工作。

2. 本书主要内容

在内容上与第 1 版相比,有了很大改变。对原来的内容进行重新整理,添加了新的例子。

本书内容可以分为三大部分:

1) 第一部分为客户端 Web 页编程技术(第 1～4 章)

第 1 章是 Web 开发技术概述,介绍了 Web 的基本概念和 Web 页的基本概念,引导读者进入网站的世界。

第 2 章介绍了超文本标记语言——HTML,它是构成网页的基础,如果要了解网站与网页编程技术,一定要掌握这部分内容。

第 3 章介绍了脚本语言——JavaScript,使用 JavaScript 可以编写出具有客户端动态功能的页面。

第 4 章介绍了层叠样式表——CSS,使用样式表可以通过简单的方式来设计网站内所有页面统一的字体、表格、背景等的显示风格,并通过一些综合实例说明前面几章所介绍的内容。

2) 第二部分为服务器端动态网页的编程技术(第 5～8 章)

第 5 章介绍了 Java 服务网页——JSP,主要介绍了 JSP 的基本语法、几个重要隐含对象的属性和方法。

第 6 章介绍了使用 JSP 访问数据库的方法,将原来 MySQL 数据库更改为使用 Access 数据库、介绍了 JSP 如何和 Access 数据库进行连接,如何在客户端对服务器端的数据库内容进行搜索、查询、编辑、删除等操作。

第 7、8 章的内容,是新添加的内容,可作为选学与自学的资料。

第 7 章介绍可扩展标记语言——XML,介绍如何通过 JSP 获取 XML 中的数据,XML 的作用。

第 8 章介绍 MVC 模式的 Java Web 应用程序,介绍如何按照 MVC 模式编写 Java Web 应用程序。

3) 第三部分为综合应用的内容(第 9、10 章)

第 9 章介绍网站开发各阶段的任务,给出网站开发过程各个阶段的工作任务,介绍网站开发的基本方法,以及网站开发规划、网站开发设计、网站开发实施等阶段的具体任务、主要成果及要求。

第 10 章介绍网站开发应用实例,以"网上体育用品商店"网站开发为例,介绍了制作网站公用程序文件、网站首页和模块化主页的方法,以及一级栏目页面与二级相关页面,动态功能模块的开发方法,可以较完整地通过实例了解开发一个网站的实现过程。

在本书中使用了 JSP 动态网页的编程技术,虽然 JSP 使用的是 Java 语言,但本书对于没有 Java 语言基础的读者同样适用,书中对所使用的 Java 语言内容都做了介绍。当然,如果学习过 Java 语言,使用本书会更加方便。在学习过程中,也可以考虑先学习 Java 语言,再学习本书介绍的 JSP 编程技术。

学习网站编程技术最好的方法是亲自动手编写相关的网页，为了使读者更方便地掌握网站编程技术，本书采用程序实例分析法，通过一系列的程序实例让读者在实践中学习网站编程技术的概念和方法。本书中所有的程序实例均通过测试，可在安装了浏览器及相关服务器软件的计算机上运行。读者只需在网上下载相关软件（使用的所有相关软件都可以在网上免费下载），进行相关的路径配置，打开一个文本编辑器，例如：Windows 操作系统“附件”里的“记事本”、“写字板” 或 Word，也可使用专门的网页编程工具，然后参照书中的介绍即可编写出一个个 HTML 和 JSP 程序。

这本书只能带领读者走进网站编程技术的大门，网站编程技术随着Internet的飞速发展也在不断地更新和发展，制作网页的工具也会越来越多，使用的网站开发设计语言也会越来越多，本书将着眼于培养网站编程技术的基本思想，掌握一定的网站编程技术，为进一步学习打下基础。

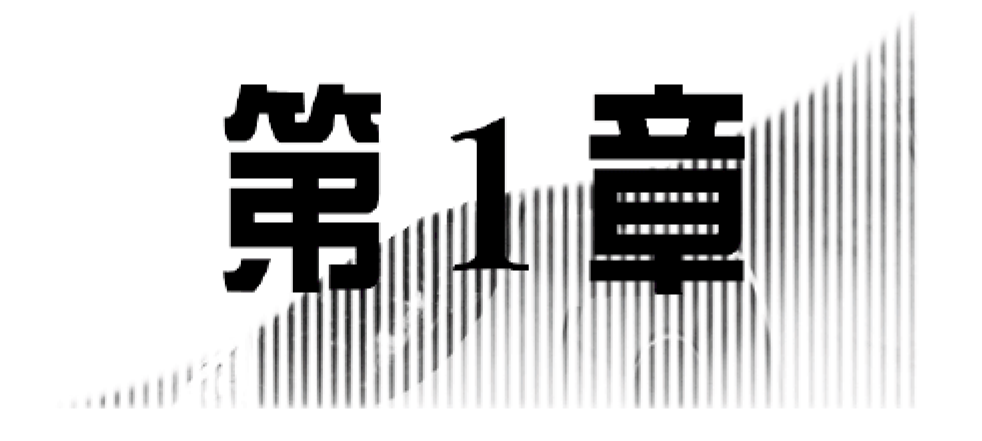
本书由邵丽萍统一编写提纲及统稿，并编写了第 6～8 章，第 4、5 章由张后扬编写，第 1、2 章由郭春芳编写，第 3 章由北京交通大学张驰编写，第 9 章由李静编写，第 10 章由山西财经大学张巨通编写。

本书有教师配套使用的电子课件及书中实例的源代码，由出版社提供给使用本教材的授课教师。

作　者

2009 年 3 月


目录



Web 开发技术概述

有过上网经历的人就到过 Web 网站，Web 网站是什么组成的呢？开发一个 Web 网站需要掌握哪些技术呢？这就是本书要介绍的内容。本书将引导读者发现一个又一个的问题，然后寻找解决问题的方案，通过实现方案解决问题，同时掌握相关的知识。

本章主要解决以下问题：

- 什么是 Web；
- 什么是 Web 网站；
- 什么是静态网页与动态网页；
- 有哪些常用的 Web 开发技术；
- 有哪些常用的编写 Web 页的工具。

1.1 什么是 Web

本节主要介绍 Web 的来历、Web 网站的构成、Web 页的特点、Web 的工作方式。

1.1.1 Web 的定义

Web 是英文 World Wide Web 的简称，意思是布满世界的蜘蛛网，中文名称为万维网。Web 由遍布在 Internet 上的称为 Web 服务器的计算机组成，它将世界各地的各种信息资源有机地组织起来，构成一个统一的 Web 网络，它是 Internet 的主干网。

Web 还是基于超文本方式、具有友好用户界面的信息查询工具和信息发布平台，通过 Web 浏览器显示 Web 页，通过 Web 页面中的超链接(Hyperlink)可以轻松的从一个 Web 页转移到其他 Web 页。这些 Web 页可以来自于世界各地的 Web 服务器，由此可见，Web 又是一个庞大的信息库。

Web 平台可以提供多种 Web 服务，例如可以提供双向信息，一种是通过浏览器浏览 Web 页面中的信息，一种是用户自己通过 Web 页面对外发布的信息。此外，在 Web 平台上世界各地的人员还可以直接进行交谈，讨论问题，还可以为商家发布广告，进行电子商务活动。Web 平台现在提供的服务越来越多。

Web 还简称为 3W、W3、WWW，中文名字还有环球网、全球信息网等。

Web 是 Web 网络、Web 服务器、Web 浏览器、Web 页的总称。Web 网络是连接世界各地 Web 服务器的网络，Web 服务器是指提供 Web 服务与存放 Web 页的计算机，Web 浏览器是安装在计算机上用来显示 Web 页的软件。Web 网络、Web 服务器、Web 浏览器是显示与使用 Web 页的基础。可见 Web 页是多么重要，接下来本书介绍 Web 页。

1.1.2 Web 页、Web 网站和主页

1. Web 页

Web 页就是上网时通过浏览器看到的一个完整页面，简称为网页。

网页可用 HTML、ASP、JSP 等语言编写，网页中可以含有标记码、文字、表格、图像、动画、超链接、声音以及视频等不同元素。每一个网页都是存放在 Web 服务器磁盘中的一个独立文件，可以单独浏览与编辑。

2. Web 网站

网页可以按一定的方式连结在一起，构成一个整体，共同描述一组完整的新闻信息，介绍部门与企业的情况，提供一组不同功能与服务。因此，将存放在 Web 服务器上具有共同主题、相似性质的一组网页及网络资源文件称为 Web 网站，简称为网站。可见，网站是由网页构成的。

3. 主页

网站总是由一个主页和若干从页组成。进入网站的第一个页面称为主页，一般将其文件命名为 index. * 。它和普通网页一样，是一个单独的网页，可以显示各种信息，但它具有特殊性，它是网站的出发点和各网页的汇总点。主页总是与一个网址（URL）相对应，可引导用户走进一个网站。在主页里，可以看到这个站点的基本信息和主要内容，使浏览的用户看到后就可知道该站点的基本内容，知道这里的信息对他是否有用，是否继续浏览，是否进行注册以得到更多的服务。因此，主页比其他网页更重要，浏览者访问 Web 网站，首先看到的就是网站的主页。图 1.1 是联想网站的主页。

4. 首页

有的网站为了让用户快速浏览到自己的网站，在进入主页前首先连接的是一个简洁的反映该网站特点的页面，有时将这种类型的主页称为首页，一般由首页提供连接到主页的路径。图 1.2 为百度网站的首页。

5. Web 页的特点

上网的人大约有 80％的时间在浏览 Web 页，因为 Web 页具有及时、新颖、美观、方便、简单和适应性强的特点。

及时、新颖体现在 Web 页显示的信息内容要是最新的。

美观是人人可见的。Web 页上不仅有文字，还有图片、颜色、动画等许多内容。



图 1.1　联想网站主页



图 1.2　百度网站首页

方便是因为 Web 页都是超文本。在 Web 页上，任何一个字符、短语、图片、按钮等都可以被指定为一个超链接，当选择访问一个超链接所指的内容时，只要单击该链接，这个内容就会自动地显示在浏览器上。超链接就像“超人”一样，跟着他可以任意地在 Web 的信息世界里遨游，所以，Web 页面又称为超文本页面。

简单主要指的是超文本描述语言 HTML，它仅仅是一组用来规定页面布局的标记且数量有限，简单易学，人人都可以快速地掌握使用 HTML 编写 Web 页的方法。

适应性强是指 Web 页独立于操作系统平台，可以用任何一种浏览器来显示，也可以用任何一种文本编辑器来编辑。

1.1.3 网址

Web 网络上有大量的 Web 服务器，服务器上有大量的 Web 页文件，如何找到需要的 Web 页呢？这就需要借助 URL 了。URL 是英文 Uniform Resource Locations 的缩写，中文称为统一资源定位器、网址、URL 地址或网站地址。URL 可以看成是一个指针，用来指定 Internet 网上一个具体的网络空间地址，它提供了一个统一的方法获取网上的信息资源。

因此，为了让其他用户在网上能看到 Web 页，在存放 Web 页与其他信息资源在 Web 服务器的时候，必须确定其网络空间地址 URL，以便浏览器能够顺利地找到这个 Web 页。有了 URL，在浏览器地址栏中输入 URL，例如 http://www.bjtu.edu.cn，不久就可以在屏幕上看到北京交通大学的网页了。

完整的网址 URL 看起来很长，例如，http://www.njtu.edu.cn/home/Homepage.htm，可将它们分为如下 4 个部分来理解：

方式://主机名/地点/文件名

1. 方式

方式指数据传输的方式，也可称为协议。关于它具体是什么内容，暂时可以不去理会，只要知道是提供的 Web 服务类型就行了。例如 HTTP、Gopher、FTP、News、Telnet、File、Mailto 等，将来还会有新增加的信息服务。HTTP 是 Web 的基本协议，是传输普通文本、声音、图像以及其他在 Internet 上可以访问的信息的协议。

2. 主机名

主机名一般指的是 Web 服务器的地址，即网页所在计算机的名字或 IP（Internet Protocol）地址、域名（Domain Name System，DNS）地址。

IP 地址由 4 部分数字组成，每部分数字不大于 256，例如 202.112.144.65。由于 IP 地址是一些数字，不好记，一般采用好记的域名地址，它由字母表示，具有一定的逻辑关系，例如 www.bjtu.edu.cn。

域名地址通常也分为四部分：机器名.单位名.单位类别.国家简称。例如，北京交通大学 Web 服务器域名地址为：www.bjtu.edu.cn。www 表示服务器名称，bjtu 表示单位名（北京交通大学），edu 表示单位类别为教育部门，cn 为中国的简称。其中机器名和单位名可以自由确定，单位类别、国家简称有一定的规则。例如，单位类别 gov 表示政府部门，mil 表示军事部门，net 表示计算机网络服务机构，com 表示商业机构，int 表示国际组织。国家简称中 uk 表示英国、ca 表示加拿大等。

注意：Internet 上的 IP 地址或域名地址需要到相关管理部门进行注册登记，以免重

复使用。

3. 地点

地点指的是在 Web 服务器上存放 Web 页的目录(/JSP/)。

4. 文件名

文件名指的是 Web 页的名字，例如，1.jsp。

根据 URL 的结构可知，http://zhc/JSP/1.htm 表示是用 http 协议访问 zhc 服务器上/JSP/目录下的 1.htm 文件，这是一个 Web 页。

不过，URL 地址通常不必包括 4 个部分，只要包含方式与主机名即可，例如 http://www.sina.com 就是一个有效的网址。当不指定文件名时，大多数的 Web 服务器都会使用一个诸如 default.* 或者 index.* 之类的文件作为默认的该服务器的主页。

除了在浏览器地址栏输入 URL 查找 Web 上的网页外，通过网页文件中的超链接也可以查找 Web 上的网页，因为每个超链接都对应一个 URL，当访问者在浏览网页时，单击超链接浏览器就会根据 URL 在 Web 上查找并显示这个网页。

1.1.4 Web 的工作方式

Web 是如何将存放在 Web 服务器中的 Web 页显示在用户的浏览器中呢?

Web 是以客户/服务器模式完成上述工作的。什么是客户/服务器模式呢?

1. 客户/服务器模式

客户是指用来与数据提供者(服务器)通信的计算机，也称为客户机，客户机可以使用服务器上的资源。客户机可由遍布世界各地的企业、单位、家庭、个人等用户使用的计算机组成，客户机与服务器相连，客户机上的浏览器可以发送或接收信息。

服务器是指能向许多客户同时提供信息资源的计算机，可由遍布世界各地的大型机构或个人的计算机构成。服务器既可以是物理的计算机，也可以是服务器软件，或者是在计算机上运行的端口监督程序。端口监督程序时刻监听着来自客户端的请求，然后交由服务器中相应程序处理请求，最后将处理后的结果返回给请求的客户。客户机和服务器可以是同一台计算机，一般是由网络连接的不同计算机分别担任客户机与服务器的任务。

如果使用过银行的 ATM 提款机，就可以很容易理解客户/服务器模式的应用。在客户机 ATM 上输入用户名和密码，提交给 ATM 连接的服务器进行验证，服务器验证无误后将结果传递给 ATM 机，用户便可在 ATM 上获得自己的账户信息并可进行提款工作。

2. 实现客户/服务器模式的三种构件

Web 要实现客户/服务器工作方式需要三种构件协调一致进行共同工作。

第一种构件是客户机上的浏览器软件，它负责显示网页内容或发送/接收信息，常用的浏览器软件有 IE、Netscape 等。

第二种构件是 Web 服务器与服务器管理软件，提供 Web 服务的计算机(软件)用来

为客户端提供信息服务，目前广泛使用的 Web 服务器管理软件有运行于 Windows 平台支持 ASP 的服务器软件 IIS(Internet Information Server)、运行于 UNIX、Linux 平台支持 PHP 的服务器软件 Apache、可在多种平台下运行支持 JSP 的服务器软件 TOMCAT。

第三种构件是 HTTP(Hypertext Transfer Protocol)，它用来在 Internet 上传送超文本协议，简称为超文本传送协议，客户机与服务器根据这个协议来传送信息。

协议(protocol)是关于信息格式及信息交换规则的正式描述。在信息技术中，协议就是一些特殊的规则集合，它被通信的接收方和发送方认可，接收到的信息和发送的信息均以这种规则加以解释。协议就好比每个国家都有自己特定的对外交流准则和交流方式，在 Internet 上，它统一了人们在网上的交流方式，可以使浏览器更加高效，使网络传输量减少。在 Web 服务器上除了 Web 页以外，还有一个 HTTP 驻留程序软件——端口监督程序，专门用于响应用户请求。

3. Web 的工作过程

Web 的工作从 Web 网络上客户端开始，客户端通过 Web 浏览器向 Web 服务器发送一个查询请求，即当用户在浏览器中输入了一个开始文件或单击了一个超级链接时，浏览器就向服务器发送一个 HTTP 请求，此请求被送往由 IP 地址指定的 Web 服务器。服务器上的 HTTP 驻留程序接收到请求后立即进行必要的操作，然后使用 HTTP 协议约定好的格式回送所要求的文件或结果信息。客户端的用户可以在浏览器上看到服务器回送的结果，如图 1.3 所示。

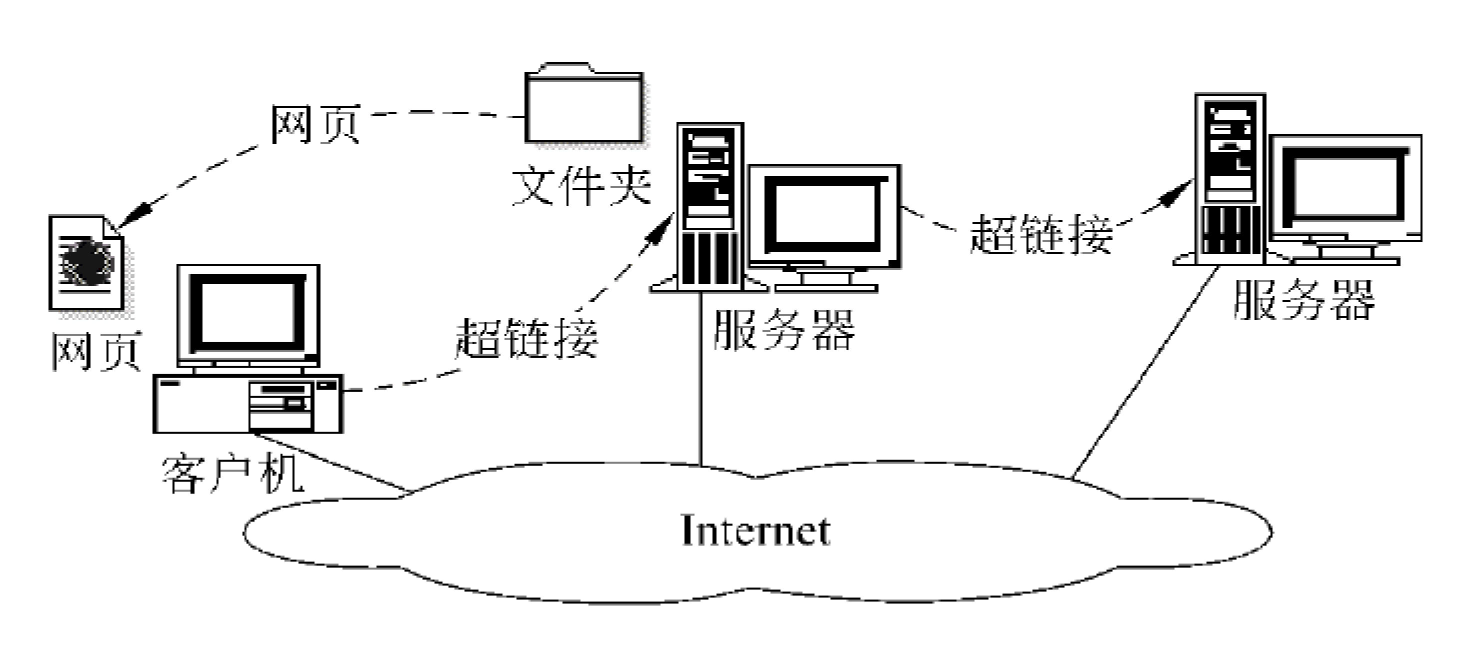


图 1.3 Web 的工作过程

由此可见，Web 服务器负责对来自客户机的请求做出回答，并且负责管理信息、寻找信息和传递信息。一个 Web 服务器除了提供它自身的独特信息外，还“指引”着存放在其他 Web 服务器上的信息。那些 Web 服务器又指向更多的 Web 服务器，当然，还可以返回到初始的 Web 服务器。如此这样，一个世界范围的信息服务器交织而成的信息网就形成了，它像蜘蛛网一样还会越来越大，故称 World Wide Web 是布满世界的蜘蛛网。

1.1.5 静态网页与请求/回答模式

使用 HTML 语言直接书写的、内容“固定不变”的网页称为静态网页。静态网页保存在 Web 服务器上，当用户浏览器通过 URL 或包含 URL 的超链接元素向 Web 服务器请求网页内容时，Web 服务器仅仅是将原已设计好的静态 HTML 文档传送给用户浏览器(如图 1.4 所示)。

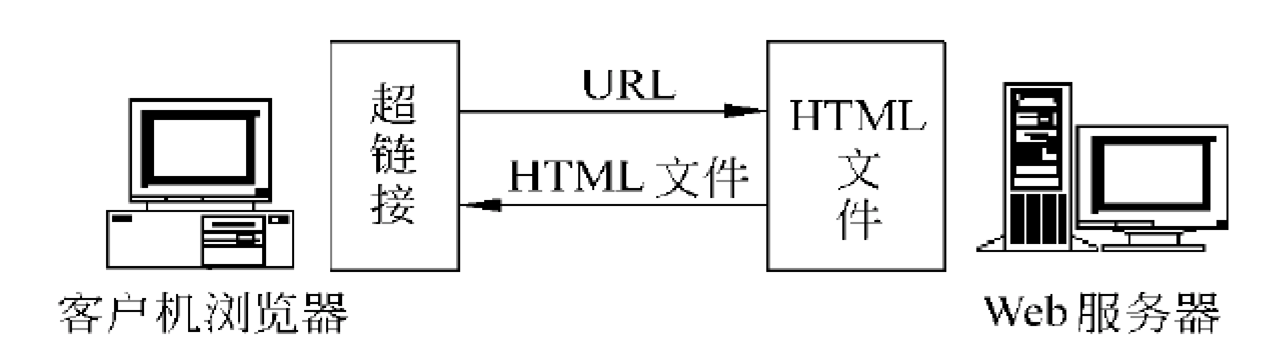


图 1.4　请求/回答模式

这种 Web 的工作模式称为 B/S(Browse/Server)结构，也可称为请求/回答模式，按照这种模式 Web 服务器与浏览器之间进行交互的步骤可归纳如下：

用户在浏览器输入 URL，例如 http://www.sina.com，按回车键后就开始对一个服务器发出 HTTP 请求。

浏览器请求域名服务器 DNS 解析 http://www.sina.com 的 IP 地址，并得到该域名的 IP 地址。

浏览器向该 IP 地址的 80 端口请求一个 TCP(Transmission Control Protocol)连接，一旦 TCP 连接成功，浏览器将发送请求信息，例如 get/index.html。

Web 服务器接收到请求信息后，首先将 Web 页 index.html 发送给浏览器，然后关闭服务器。

浏览器将显示 Web 页 index.html 文件中的文档内容。

1.1.6　动态网页与三层 B/S 模式

内容能够因人因时变化、能够在客户端与服务器端进行交互的网页称为动态网页。在动态网页中，常用到 Web 数据库的数据，图 1.5 为引入 Web 数据库后的三层 B/S 模式示意图。

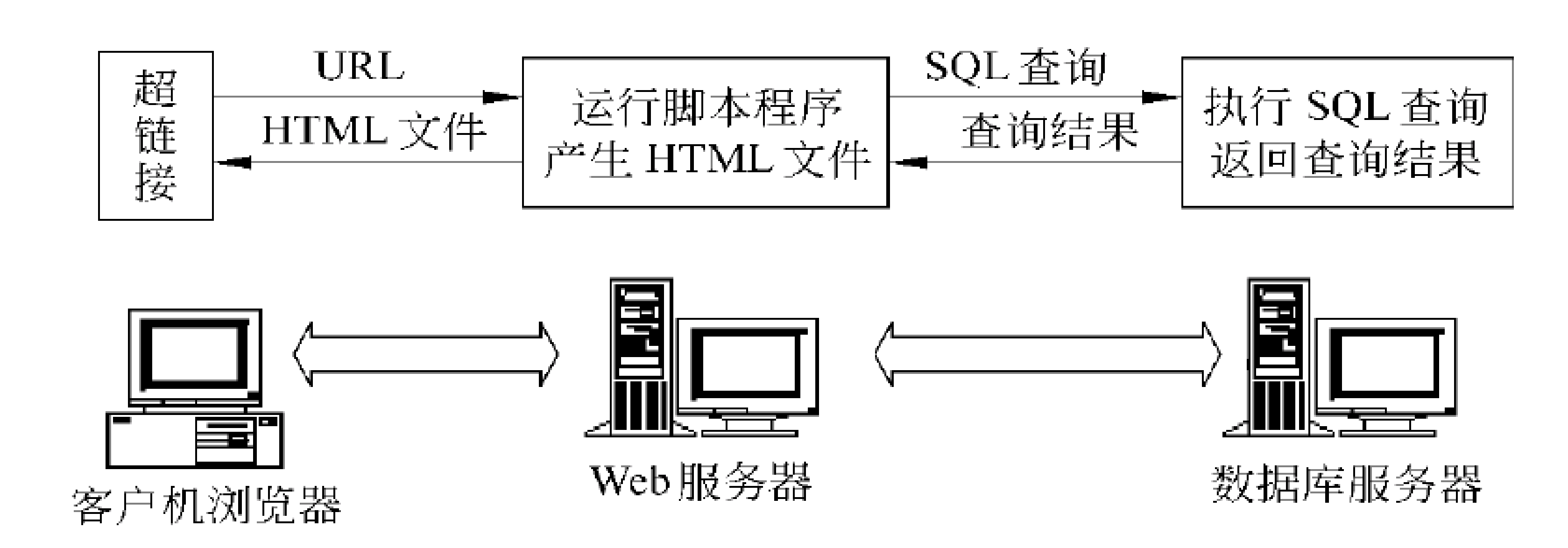


图 1.5　三层 B/S 模式

当客户机的浏览器通过 URL 向 Web 服务器请求下载信息时，在 Web 服务器上要运行嵌入在网页中的脚本程序，脚本程序通过 SQL 查询调用数据库服务器中存储的数据，数据库服务器执行查询操作，将结果返回到 Web 服务器，产生特定格式的 HTML 文件。这样当客户端浏览器通过 URL 请求信息到显示信息完毕，客户端接收到的 HTML 文件中的内容实际上是由数据库提供的。这种方式使处理更灵活，自由度更大。

出于效率考虑，Web 服务器和数据库服务器常在一个局域网中，甚至在同一台机器上。由动态网页组成的网站称为动态网站，它有如下优点。

(1) 自动更新。即无需手动更新 HTML 文档，便会自动生成新的页面，可以大大节

省工作量，例如，新闻网站。

（2）交互性。即网页会根据用户的要求及选择而动态改变和响应，将浏览器作为客户端界面。例如电子商务中的网上超市、网上书店、留言簿等，这将是今后 Web 发展的趋势。

（3）因时因人而变。即当不同的时间、不同的人访问同一网址时会产生不同的页面，例如，网上教务系统。

1.2 Web 开发技术的起源——HTML

Web 网络的历史很短，但它已经从一个大部分显示静态信息的网络媒体演化到能对股票进行交易和进行网上购书等操作的一个基础设施。在 Web 网络上可以使用各种各样的应用程序，这些应用程序组成的信息系统进一步扩大了 Web 的应用。它不仅表现在网络上，还表现在企业及其他各个领域，以至传统的基于客户机/服务器的应用程序都在逐渐被基于浏览器/服务器的应用程序所替代。

基于浏览器/服务器的应用程序较传统的基于客户机/服务器的应用程序有很多优点，这些优点包括几乎没有限制的客户端访问和极为简化的应用程序部署和管理（要更新一个应用程序，管理人员只需要更改一个基于服务器的程序，而不是成千上万的安装在客户端的应用程序）。这些变化使得软件产业迅速地向 B/S 模式的多层次应用程序迈进。

B/S 模式多层次应用程序快速增长的需求，使得 Web 开发技术不断改进。HTML 对于显示相对静态的内容是不错的选择，但不能创建交互的 Web 页，交互的页面内容是基于用户的请求或者系统的状态来确定的，不是预先定义的文字。为满足这种需要，一些动态的 Web 开发技术应运而生。因此，网上不仅可以看到后缀名为.html 或.htm 的网页，还能看到很多后缀名为.jsp、.asp、.aspx、.php、.phpx 的 Web 页，这些 Web 页是不同的，它们表示网页是由不同的 Web 开发技术编写的文件。

本节的内容主要介绍 Web 开发技术的起源 HTML 语言的产生、发展、构成与不足。

1.2.1 HTML 的发展过程

作为一种实用的超文本语言，HTML 的历史最早可以追溯到 20 世纪 40 年代。1945 年，Vannevar Bush 在一篇文章中阐述了文本和文本之间通过超级链接相互关联的思想，并在文中给出了一种能实现信息关联的计算机 Memex 的设计方案。Doug Engelbart 等人则在 1960 年前后，对信息关联技术做了最早的实验。与此同时，Ted Nelson 正式将这种信息关联技术命名为超文本（hypertext）技术。1969 年，IBM 的 Charles Goldfarb 发明了可用于描述超文本信息的 GML（Generalized Markup Language，通用标记语言）。1978—1986 年间，在 ANSI（美国国家标准协会）等组织的努力下，GML 语言进一步发展成为著名的 SGML（Standardized Generalized Markup Language，标准通用标记语言）语言标准。当 Berners Lee（伯纳斯·李）和他的同事们在 1989 年试图创建一个基于超文本的分布式应用系统时，他意识到 SGML 是描述超文本信息的一个上佳方案，美中不足的是 SGML 过于复杂，不利于信息的传递和解析。于是，伯纳斯·李对 SGML 语言做了大刀阔斧的

简化和完善，设计了一种为 Web 度身定制的语言——HTML（Hyper Text Markup Language，超文本标记语言）来展现超文本信息。

HTML 的设计思想非常简单，伯纳斯·李只是希望在网上显示文字并通过页面的超链接实现网页的连接，他设计了一个初级浏览器和一个编辑系统并在网上合二为一，通过它们来显示 HTML 编写的超文本信息，通过浏览器实现超链接。1990 年 11 月，第一个 Web 服务器 nxoc01. cern. ch 开始运行，伯纳斯·李在自己编写的图形化 Web 浏览器 World Wide Web 上看到了最早的 Web 页面。

由于伯纳斯·李设计的 HTML 是以文本格式为基础的，可以用任何编辑器和文字处理器创建，而且它仅有不多的几个标记（tag），所以很容易掌握编写 HTML 文档的方法。HTML 很快在 Web 上得到普及与使用。HTML 使 Web 网络开始了新篇章，Web 网络因此得到迅猛发展，在 Web 网可以很容易地发布信息与浏览文本信息。1991 年，CERN（European Particle Physics Laboratory，欧洲粒子物理实验室，万维网的发祥地）正式发布了 Web 技术标准。目前，与 Web 相关的各种技术标准都由著名的 W3C（World Wide Web Consortium，万维网协会）组织管理和维护，W3C 的主任是伯纳斯·李。因为在互联网技术上的杰出贡献，伯纳斯·李被业界公认为“互联网之父”。他的发明改变了全球信息化的传统模式，带来了一个信息交流的全新时代。然而比他的发明更伟大的是，伯纳斯·李（见图 1.6）并没有像其他人那样为 WWW 申请专利或限制它的使用，而是无偿地向全世界开放。



图 1.6　伯纳斯·李

能在网上看到文本后，人们又提出了新的问题：能否在网上看到图像、动画呢？早期的浏览器仅能显示文本文字。如何解决这个问题呢？1993 年，一个名叫 Marc Andreessen 的大学生在他的 Mosaic 浏览器上加入了图像<img>标记，从此在 Web 网络上可以看到图像了。而后，Mosaic 浏览器随<img>推向市场，伯纳斯·李加入了初期的国际互联网研究组织，而 Marc 在加利福尼亚创建了一个名为 Netscape 的浏览器公司。

但人们认为仅有图像和图标还远远不够，希望可以将任何形式的媒介信息加到网页上去。因此，HTML 不停地发展，不断产生新型、功能强大且生动有趣的标记形式和可以识别它们的浏览器。所以，出现了<background>（背景）、<frame>（框架）、<font>（字体）和<blink>（闪烁效果）等标记。Microsoft 公司致力于网上游戏领域，他们设计了<marquee>（滚动的字）和<bgsound>（背景声效）等标记。

越来越多的标记会不会给浏览器造成困惑呢？HTML 的一个优越特性是如果一个 Web 浏览器不能识别一个标记属性时，这个属性将被忽略，而且不会有任何副作用。所以，在 HTML 中可以放心地使用各种标记。

HTML 的发展过程很有启发意义，而且由于 HTML 的出现，使互联网络得到迅猛地发展，目前人们的工作与生活已经越来越离不开网络了。

1.2.2 HTML 文档的组成

用 HTML 编写的超文本文档称为 HTML 文档，它能独立于各种操作系统平台(如 UNIX，Windows 等)。自 1990 年以来 HTML 一直被作为 Web 上的信息表示语言，用于描述页面的格式设计和它与 Web 上其他页面的连接信息。

HTML 语言是由描述性的标记符(称为标记码)构成的。标记码是区分文本各个组成部分的分界符，它把 HTML 文档划分成不同的结构部分，例如页面、标题、段落、正文主体、超链接、表格、图像、表单等，并向浏览器提供该文档的格式化信息，以显示文档的外观特征。标记码由一对尖括号和英文字母及其他符号构成，例如<html>。

HTML 文档包含两种信息，一种是页面要显示的各种文本文字；一种是标记码，用来描述页面的不同元素、结构、文本格式和超文本链接。HTML 文档是一个 ASCII 文本文件，通常它带有.html 或.htm 后缀名。

例 1.1 一个简单的 HTML 文档(文件名为 1-1.html)。

```
<html>
<head>
<title>
    Web 页面的标题
</title>
</head>
<body>
HTML 语言的范例：
  <p>
<img src="girl.gif" height=120 width=90>
</p>
</body>
</html>
```

1-1.html 文档通过标记码描述了显示文本文字“HTML 语言的范例”和一个图像元素 girl.gif 的格式。在浏览器里浏览该页面时，浏览器将把每个标记码解释成它指明的含义，在默认位置以指定的样式将文字与图像显示在屏幕上。在屏幕上看不到 HTML 标记码，看到的只有要显示的文字和图像，如图 1.7 所示。



图 1.7 HTML 页面显示的文字与图片

通过 HTML 的标记码，可以设置文本文字的字体和颜色，插入各种图像和表格，还可以加入指向 Internet 上任意一个 Web 页的超链接。正是有了这些 HTML 标记码，Web 页才变得漂亮美观，多姿多彩。HTML 语言使网络

从一个静态的文本世界变成了一个丰富多彩的多媒体世界。HTML 是网上最早的公用语言。

1.2.3 HTML 的不足

随着 Web 应用的发展，HTML 语言的不足开始显现出来。主要表现如下：

1. 静态性

由于 HTML 文档在浏览时不会因时因地而发生变化，也不允许在浏览器加载后更改页面内容，所以 HTML 文档为静态的、固定的，它限制了其更新的速度，如果要更新其内容，整个页面的内容需要全部修改。

2. 格式与布局的局限

在 HTML 文档中不能随意扩展文本格式。不能设计像报纸、杂志那样漂亮的布局。

3. 不能设计计算功能

在 HTML 文档中不能实现高级程序设计语言的计算功能，所以不能满足人们对网络越来越多的功能需求。

HTML 的不足给其他 Web 开发技术留下了空间，一些新的、动态的 Web 开发技术开始出现。

1.3 客户端技术

Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及到客户端和服务端两个层面。因此，Web 开发技术大体上也可以被分为客户端技术和服务端技术两大类。

本节的内容主要介绍 Web 客户端技术中的典型代表 JavaScript、VBScript、CSS 和 DHTML。

1.3.1 JavaScript 和 VBScript

Web 客户端的主要任务是展现信息内容，而 HTML 语言则是信息展现的最有效的载体之一。

为了弥补 HTML 静态性的不足，在 HTML 页面中引入了 GIF 图像格式文件，使 HTML 页面第一次有了动感元素。但更大的变革来源于 1995 年 Java 语言的问世。Java 语言天生具备平台无关的特点，可在浏览器中开发动态应用。1996 年，著名的 Netscape 浏览器在其 2.0 版中增加了对 Java Applets 和 JavaScript 的支持。Microsoft 的 IE 3.0 也在这一年开始支持 Java 技术。从此，喜欢动画、喜欢交互操作、喜欢客户端应用的开发人员可以用 Java 或 JavaScript 语言随心所欲地丰富 HTML 页面的计算功能了。

JavaScript 语言在所有客户端开发技术中占有非常独特的地位：它是一种以脚本方

式运行的，简化了的 Java 语言，这也是脚本技术第一次在 Web 世界里崭露头角。为了用纯 Microsoft 的技术与 JavaScript 抗衡，Microsoft 在 1996 年为 IE 3.0 设计了另一种后来也声名显赫的脚本语言——VBScript。

JavaScript 和 VBScript 是脚本语言的典型代表。脚本语言是指介于 HTML 和诸如 Java、Visual Basic、C++ 等编程语言之间的一种特殊的语言，尽管它更接近后者，但它没有编程语言复杂、严谨的语法和规则。它们可以嵌入 HTML 文档之中，与 HTML 语言一起构成 HTML 文档。

脚本语言可以实现简单的计算功能。因为脚本是由一系列的程序语言命令语句组成，如同一般的程序语言，脚本可以定义变量并赋值给变量，通过一系列命令定义一个过程或函数来完成一个功能计算。例如，通过 JavaScript 可以在网页上输入计算信息，不用任何的网络来回传送，即不用传给服务器(server)处理，浏览器通过脚本语言，可直接将计算的结果显示在页面上。计算功能使 HTML 文档具有了动态特征。

1.3.2 CSS 与 DHTML

通过客户端让 HTML 页面实现动态特征的还有 CSS(Cascading Style Sheets)和 DHTML(Dynamic HTML)技术。

1. CSS

CSS 中文称为层叠样式表单，简称为样式表。顾名思义，它是一种设计网页样式的技术。使用 CSS 技术，可以有效地对页面的布局、字体、颜色、背景和其他效果实现更加精确的控制。1996 年底，W3C 提出了 CSS 的建议标准，同年，IE 3.0 引入了对 CSS 的支持。CSS 大大提高了开发者对信息展现格式的控制能力。1997 年的 Netscape 4.0 不但支持 CSS，而且增加了许多 Netscape 公司自定义的动态 HTML 标记，这些标记在 CSS 的基础上，让 HTML 页面中的各种要素"活动"了起来。

2. DHTML

1997 年，Microsoft 发布了 IE 4.0，并将动态 HTML 标记、CSS 和动态对象模型(DHTML Object Model)发展成了一套完整、实用、高效的客户端开发技术体系，Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果，DHTML 技术无需启动 Java 虚拟机或其他脚本环境，可以在浏览器的支持下，获得更好的展现效果和更高的执行效率。

DHTML 最显著的特点是完全客户端的技术，它可以直接通过浏览器中的 Web 页实现页面与用户之间的交互。与 Java、Flash 等技术不同的是，用 DHTML 编写的页面不需要插件的支持就能完整地实现其动态功能。

DHTML 集成了 HTML、CSS、JavaScript 和 VBScript 的功能。它通过 CSS 样式表可以重新定义 Web 页面的显示格式，弥补了 HTML 格式与布局的局限性，使页面布局具有更大的灵活性。它通过 JavaScript 和 VBScript 脚本语言增强页面的应用功能，改变网页的内容，在 Web 页面直接建立动画、游戏和计算功能。通过 DHTML 的数据绑定技术

可以将数据从数据源传递到 Web 页中，实现从服务器上动态地返回数据。

DHTML 可以说是 Web 技术发展到一定阶段，适应人们迫切需求的一种必然技术。虽然，随着 ASP、JSP 的出现，DHTML 的许多功能逐渐被取代。但是，ASP、JSP 是基于服务器端执行的，而 DHTML 是在客户端执行的，因此将二者结合会得到更加完美的效果。

1.3.3 插件

为了在 HTML 页面中实现音频、视频等更为复杂的多媒体应用，1996 年的 Netscape 2.0 成功地引入了对 QuickTime 插件的支持，从此，插件这种开发方式迅速风靡了浏览器的世界。

在 Windows 平台上，Microsoft 将客户端应用集成放在了 ActiveX（动态控件）身上。1996 年，IE 3.0 正式支持在 HTML 页面中插入 ActiveX 控件的功能，这为其他厂商扩展 Web 客户端的信息展现方式开辟了一条自由之路。1999 年，Realplayer 插件先后在 Netscape 和 IE 浏览器中取得了成功，与此同时，Microsoft 自己的媒体播放插件 Media Player 也被预装到了各种 Windows 版本之中。

最重要的是 Flash 插件的问世：1990 年代初期，Jonathan Gay 在 Future Wave 公司开发了一种名为 Future Splash Animator 的二维矢量动画展示工具，1996 年，Macromedia 公司收购了 Future Wave，并将 Jonathan Gay 的发明改名为 Flash。从此，Flash 动画成了 Web 开发者表现自我、展示个性的最佳方式。

目前在浏览器中使用插件可以显示不同格式的信息元素。

1.4 服务端技术

Web 服务端的开发技术是在解决静态网页维护难（文件数目多）、查询数据难、修改页面难的缺陷中产生与发展起来的。

服务端技术中最重要的是动态网页开发技术，即利用后台数据库服务器中的数据动态生成超文本。要改变客户端 Web 页的式样，只需修改服务端几个 CSS 文件；要变更客户端 Web 页里的内容，只需对服务器中的数据库进行数据操作，不仅更新速度快而且可以保持原有的页面风格不变；对于全文检索等查询应用，服务端技术的优越性更为突出。

通常所说的动态也包括图片的动态显示，例如动态 GIF、Java Applets、flash 动画等，本书所指的动态主要指页面中资料和数据的动态性，包括如下几方面内容。

1. 交互性

客户端的网页会根据用户的要求与选择而动态改变、产生相应数据。

2. 自动更新

无需手动更新客户端的 HTML 文档，它能自动生成新的页面。

3. 因时因人而变

当不同的时间、不同的人访问同一网址时会产生不同的页面。

本节主要介绍 Web 服务端技术中的典型代表 CGI、PHP、ASP 和 JSP。

1.4.1 CGI

最早的 Web 服务器只是简单地响应浏览器发来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器。后来出现了一种名为 SSI(Server Side Includes)的技术可以让 Web 服务器在返回 HTML 文件前，更新 HTML 文件的某些内容，但其功能非常有限。

第一种真正使服务器能根据运行时的具体情况，动态生成 HTML 页面的技术是 CGI(Common Gateway Interface)技术，中文称为网关接口。1993 年，CGI 1.0 的标准草案由 NCSA 提出，1995 年，NCSA 开始制定 CGI 1.1 标准，1997 年，CGI 1.2 也被纳入了议事日程。CGI 技术允许服务端的应用程序根据客户端的请求，动态生成 HTML 页面，这使客户端和服务端的动态信息交换成为了可能。随着 CGI 技术的普及，聊天室、论坛、电子商务、信息查询、全文检索等各式各样的 Web 应用蓬勃兴起，人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。Web 开始有了动态网页开发技术。

不过 CGI 比较复杂，学起来有些困难，不仅要懂得 CGI 的工作原理，还要是个编程高手。因此，出现了 PHP、ASP、JSP 等服务器端的动态网页开发技术。

1.4.2 PHP

PHP 是英文 Hypertext PreProcessor 或 Personal Home Page Tools 的缩写，中文称为超级文本预处理语言。与 CGI 程序不同，PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面，Web 应用的开发者可以用一种更加简便、快捷的方式实现 Web 的动态功能。

PHP 是 Rasmus Lerdorf 在 1994 年秋天构思出来的，开始的目的只是用来与浏览他主页在线简历的人保持联系。第一个交付用户使用的版本是在 1995 年初发行的，初始时它仅包括一个懂得几条宏指令的分析引擎和一组用于主页信息反馈的工具(一个留言簿与一个计数器)。1995 年年中，Rasmus 重写了这个工具软件，并取名为 PHP/FI。FI 来源于他写的另外一个 HTML 表单集成数据的软件包。他把个人主页工具与表单集成工具合并在一起，并加入了对 MySQL 数据库的支持程序，构成了 PHP/FI。此后，PHP/FI 便以一种令人惊奇的速度传播开来，人们开始大量使用它编写程序。

无法确切地统计它的使用范围，根据估计在 1996 年底，世界上已经有至少 15000 个站点在使用 PHP/FI。1997 年年终这个数字已经增长到超过 50000 个站点。1997 年年底，PHP 的发展出现了一个重大的转折，这就是 PHP 的开发从 Rasmus 个人的爱好升级到一群程序员们有组织的工作。再后来，这个工具软件被 Zeev Suraski 和 Andi Gutmans 重写，通过这次全面的重写，大量 PHP/FI 的功能被移植到 PHP3 中，并且成为了 PHP

3.0 的基本雏形。到 1998 年年底,已经有大量的商业化产品,根据 NetCraft 网站估计的保守数据,PHP 已经被世界上 150000 个站点所采用。在网上经常可以看到以.php 作为后缀名的 Web 页。

PHP 有以下特性。

(1) 开放的源代码:所有的 PHP 源代码事实上都可以得到。

(2) 免费 :PHP 是一种免费软件,常与免费的 Web 服务器软件 Apache 和免费数据库软件 MySQL 配合使用于 Linux 平台上,具有较高的性能价格比。

(3) 基于服务器端:由于 PHP 是运行在服务器端的,即使再多的用户,再复杂的程序也不会影响运行的速度。

(4) 跨平台:PHP 可以运行在 UNIX、Linux、Windows 环境。

(5) 嵌入 HTML:PHP 语言被嵌入在 HTML 中,容易理解。

(6) 简单的语言:PHP 坚持以脚本语言为主,与 Java 和C++ 标准的程序设计语言不同,容易学习。

(7) 效率高:PHP 消耗相当少的系统资源。

(8) 图像处理:用 PHP 可以动态创建图像。

(9) 强大的数据库支持:目前其支持范围覆盖了包括 Oracle、Sybase、Microsoft SQL、MySQL、Informix、Solid dBase、ODBC、UNIX dbm、PostgreSQL、Adabas D 等在内的大多数常见数据库。使用它编写一个含有数据库功能的网页程序十分简单。

1.4.3 ASP

1996 年,Microsoft 借鉴 PHP 的思想,在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。ASP 是英文 Active Server Page 的缩写,中文称为动态服务网页。ASP 使用的脚本语言是 VBScript 和 JavaScript。借助 Microsoft Visual Studio 等开发工具在市场上的成功,ASP 迅速成为了 Windows 系统下 Web 服务端的主流开发技术。

ASP 与 HTML 有什么区别呢?简单地说,ASP 是对 Web 服务器功能的增强,ASP 改变了 HTML 只是浏览器从 Web 服务器下载 Web 页面的工作方式。ASP 不仅可以让客户端的浏览器从服务器取得 Web 页,还可以让 Web 服务器随时接收来自客户端浏览器发来的信息,根据接收到的用户信息,进行数据处理,修改 Web 服务器中的网页内容,再将修改后的 Web 页返回到客户端浏览器。

ASP 有以下特点:

(1) ASP 可以创建基于 Web 的应用程序。

(2) ASP 由 VBScript、JavaScript 等简单易懂的脚本语言命令、HTML 标记和 ActiveX 组件组合而成。

(3) 可使用普通的文本编辑器编写 ASP 文件,例如 Windows 的记事本、Word 等。

(4) ASP 文件与浏览器无关,客户端只要使用可浏览 HTML 语言的浏览器,即可浏览 ASP 所设计的网页。

(5) ASP 文件无须编译,其所使用的脚本语言(VBScript、JavaScript)均可以在 Web 服务器端直接执行。用户端的浏览器不需要理会这些脚本语言。

(6) 提高了程序的安全性,ASP 的源程序不会被传到客户浏览器,因而可以避免所写的源程序被他人剽窃。在客户端看到的是 ASP 页面是 ASP 文件的执行结果。

(7) 可创建具有交互功能的 Web 页面。

(8) ASP 可以与多种数据库建立连接,在 ASP 应用程序中可以方便地使用、修改、添加数据库中的数据。

1.4.4 JSP

Sun 公司并没有因为有了 JavaScript、Java Applets 技术而停止不前,1997 年,Sun 公司提供了 Servlet 技术,它是对支持 Java 的服务器的扩充。它的主要用途是扩展 Web 服务器的功能,提供安全的、可移植的、易于使用的 CGI 替代品。它是一种动态加载的模块,可以为来自 Web 服务器的请求提供服务。

1998 年,Sun 公司诞生了专门制作 Web 动态网页的技术 JSP。JSP 是英文 Java Server Pages 的缩写,中文称为 Java 服务网页。JSP 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准,其在动态网页的建设中有着强大而特别的功能。在创建动态内容页面时 JSP 提供了一个满足下列条件的解决方案。

(1) 能够在任何 Web 或应用程序服务器上运行;

(2) 可以将应用程序逻辑和页面显示分离;

(3) 能够快速地开发和测试;

(4) 能够简化开发基于 Web 的交互式应用程序的过程。

JSP 技术标准(规范)是 Web 服务器、应用服务器、交易系统、以及开发工具供应商间广泛合作的结果。在开发 JSP 技术标准(规范)的过程中,Sun 公司与许许多多主要的 Web 服务器、应用服务器和开发工具供应商,以及各种各样富有经验的开发团体进行合作,其结果是找到了一种为应用和页面开发人员使用的可移植性强、易用的 JSP 开发方法。

在 Sun 正式发布 JSP 之后,这种新的 Web 应用开发技术很快引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。

JSP 有如下特点。

1) 将内容的生成和显示进行分离

使用 JSP 技术,可以更明确地进行分工,Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面。JSP 程序员使用 JSP 标识或者脚本来设计页面上的动态内容(内容是根据请求变化的,例如请求账户信息或者特定的一瓶酒的价格将显示不同的页面)。

2) 生成可重用的组件

绝大多数 JSP 页面依赖于可重用的,跨平台的组件(例如 JavaBeans)和执行特殊任务的 Enterprise JavaBeans 组件来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件,或者使得这些组件为更多的使用者或客户团体所使用。基于组件的方法加速了总体开发过程,并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

3）采用标识简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的编程人员。JSP 技术封装了许多功能，这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成时所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件，设置或检索组件属性，下载 Applet，以及执行用其他方法更难于编码和耗时的功能。

4）可扩展性

通过开发定制化标识库，JSP 技术是可以扩展的。第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样执行特定功能的构件来工作。

5）具有 Java 技术的所有优势

由于 JSP 内置的脚本语言是 Java 编程语言，而且所有的 JSP 页面都被编译成为 Java 字节代码，所以 JSP 具有 Java 技术的所有优势，包括健壮性和安全性。所以，大型商业网站通常都使用 JSP 进行开发。

6）一次编写，各处运行

作为 Java 平台的一部分，JSP 拥有 Java 编程语言一次编写，各处运行的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中，可以任意选择服务器和工具，更改工具或服务器并不影响当前的 JSP 应用程序。

JSP 很好地结合了 Servlet 技术，它同时拥有类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能，此外，Java 的运行时编译技术也大大提高了 Servlet 和 JSP 的执行效率，Servlet 和 JSP 也是 J2EE 平台的核心技术。

1.5　企业级开发平台

服务端开发技术的日益完善使开发复杂的 Web 应用成为了可能。在电子商务活动中，为了适应企业级应用开发的各种复杂需求，为了给最终用户提供更可靠、更完善的信息服务，两个最重要的企业级开发平台——J2EE 和.NET 在 2000 年前后分别诞生于 Java 和 Windows 阵营，它们在企业级 Web 开发领域展开了激烈的竞争。这种竞争关系促使了 Web 开发技术以前所未有的速度提高和跃进。

本节的内容主要介绍企业级 Web 开发平台中的典型代表 ASP.NET 和 J2EE。

1.5.1　Microsoft 公司的.NET

Microsoft 的.NET 平台是一个强调多语言间交互的通用运行环境。2001 年，ECMA(European Computer Manufactures Association，欧洲计算机制造商协会)通过了 Microsoft 提交的 C#语言和 CLI 标准，这两个技术标准构成了.NET 平台的基础，它们于 2003 年成为了 ISO(International Organization for Standardization，国际标准化组织)的国际标准。2002 年，Microsoft 正式发布.NET Framework 和 Visual Studio .NET 开发环境。

在.NET 发布之前，有许多 Windows 平台的 Web 开发者就利用 Beta 版本开发 Web

应用了。因为.NET 平台及相关的开发环境不但为 Web 服务端应用提供了一个支持多种语言的、通用的运行平台，而且还引入了 ASP.NET 这样一种全新的 Web 开发技术。

ASP.NET 超越了 ASP 的局限，可以使用 VB.NET、C# 等编译型程序设计语言，支持 Web Form、.NET Server Control、ADO.NET 等高级特性。

Microsoft 公司推出的 ASP.NET 不是 ASP 的简单升级，而是 Microsoft 推出的新一代 Active Server Pages。ASP.NET 是.NET 的一部分。在许多方面，ASP.NET 与 ASP 有着本质的不同。ASP.NET 完全基于模块与组件，具有更好的可扩展性与可定制性，数据处理方面引入了许多新技术，正是这些具有革新意义的新特性，让 ASP.NET 远远超越了 ASP，同时也提供给 Web 开发人员更好的灵活性，有效缩短了 Web 应用程序的开发周期。

ASP.NET 有以下特点：

1）执行效率大幅提高

ASP.NET 是把基于通用语言的程序在服务器上运行，与以前的 ASP 即时解释程序不同，它是将程序在服务器端首次运行时进行编译，这样的执行效果，当然比一条一条地解释快很多。

2）有开发工具支持

ASP.NET 构架可以用 Microsoft 公司最新的产品 Visual Studio.NET 开发工具进行开发，WYSIWYG（What You See Is What You Get 所见即所得）的编辑方式可加快程序的开发过程。

3）支持程序语言

ASP.NET 现在支持 C#（C++ 和 Java 的结合体）与 VB 语言。不像 ASP 只是支持脚本语言，所以功能更强大。

4）可管理性

ASP.NET 使用一种字符基础的，分级的配置系统，使服务器环境和应用程序的设置更加简单。因为配置信息都保存在简单文本中，新的设置可能不需要启动本地的管理员工具就可以实现。一个 ASP.NET 的应用程序在一台服务器系统的安装只需简单拷贝一些必须的文件，不需要系统重新启动。

5）多处理器环境的可靠性

ASP.NET 已经被刻意设计成为一种可以用于多处理器的开发工具，它在多处理器的环境下用特殊的无缝连接技术，将大大提高运行速度。即使现在的 ASP.NET 应用软件是为一个处理器开发的，将来多处理器运行时不需要任何改变都能提高它们的效能，而现在的 ASP 却做不到这一点。

6）自定义性和可扩展性

ASP.NET 设计时考虑了让网站开发人员可以在自己的代码中自己定义 plug-in 的模块，加入自己定义的任意组件。

1.5.2 Sun 公司的 J2EE

J2EE 是英文 Java 2 Platform Enterprise Edition 的缩写，中文称为 Java 2 平台企业

版，J2EE 是纯粹基于 Java 的解决方案。

1998 年，Sun 发布了 EJB 1.0 标准。EJB(Enterprise JavaBean)是 Sun 的服务器端组件模型，最大的用处是部署分布式应用程序，凭借 Java 跨平台的优势，用 EJB 技术部署的分布式系统可以不限于特定的平台。EJB 为企业级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。

1999 年，Sun 正式发布了包含三大核心技术 Servlet、JSP 和 EJB 的 J2EE 1.0 版本。紧接着，遵循 J2EE 标准，为企业级应用提供支撑平台的各类应用服务软件开始出现，IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里最为成功的商业软件平台。到 2003 年时，Sun 的 J2EE 版本已经升级到了 1.4 版，其中三个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。至此，J2EE 体系及相关的软件产品已经成为了 Web 服务端开发的一个强有力的支撑环境。

J2EE 是使用 Java 进行企业开发的一套扩展规范，它提供基于组件设计、开发、部署和管理企业应用的解决方案。J2EE 具有许多优点，例如"一次编译、随处运行"的跨平台特性、方便高效的数据库提供无缝连接的技术 JDBC(Java Data Base Connectivity，Java 数据库连接)、能够在 Internet 应用中确保数据完整性的安全模式以及支持多线程技术等，同时还针对 Servlet、JSP、EJB、JMS、JTS/JTA、JCAT 和 XML 等技术提供了全面的支持。J2EE 已建立了一套能够使企业级应用开发者缩短软件产品开发周期的标准架构体系。

事实上，J2EE 已成为企业级开发的工业标准和主流平台。J2EE 为建立具有高可用性、高可扩充性、易维护性的企业级系统提供了完美的机制和架构模型。在 J2EE 提供的统一开发平台下，可以大大降低开发多层应用的成本和复杂性，对于需要获得高可用性、高安全性以及高可扩充性软件服务而又缺乏资金的企业来说，J2EE 的中间层集成框架完全可以胜任并满足其要求。同时，J2EE 自身具备的完备的目录服务支持、高度的安全机制、高效的性能展现等众多优势技术，为企业平滑集成已有应用系统发挥了巨大作用。

基于 J2EE 的企业级应用系统，不受任何特定操作系统、网络环境、硬件平台的束缚。应用系统可开发并部署在异构的环境中，并能确保系统的可移植性。因此，只要符合 J2EE 规范的系统开发一次即可部署到各种操作系统平台和网络平台上。J2EE 作为长期的性能稳定的企业级应用系统平台，能保障 7×24 小时的不间断服务，以满足企业客户、合作伙伴的需要，是企业理想的运行平台也是 Internet 全球化发展的客观必然要求。J2EE 高伸缩性的特点在如今"随需应变"(IBM 公司 2002 年底提出)电子商务战略大行其道的时代，做出了卓越贡献。J2EE 采用负载平衡策略，使系统软硬件资源利用率及性能表现最大化，消除了系统中的瓶颈。允许多台服务器集成部署，实现可高度伸缩的系统，在保存企业现有 IT 资产减少重复性投入的同时也满足了未来商业应用的需要。

1.6 Web 应用模型与开发框架

2000 年以后，随着 Web 应用的日益复杂，人们逐渐意识到，单纯依靠某种技术多半无法达到快速开发、快速验证和快速部署的最佳境界。研究者开始尝试着将已有的 Web

开发技术综合起来,形成完整的应用模型或开发框架,以此满足各种复杂的应用需求。

Microsoft 在客户端的应用技术集成方面走在了最前面。1998 年时 Microsoft 推出的 Windows 98 就可以在桌面上集成 Web 页面,这实际上是将资源管理器和 Web 浏览器的功能有效地结合了起来。2000 年后,Microsoft 陆续推出了 MSN Explorer 和与之相关的 MSN 在线服务。这一应用模型将 Web 浏览、视频点播、邮件处理、网上游戏、在线聊天等许多种用户常用的 Web 功能集成在了一个统一的界面中。从信息利用的角度看,MSN 试图让用户在一个最舒适的环境中获取足够的信息。另一个与客户端应用技术集成相关的例子是搜索引擎 Google 在 2003 年展示给大家的 Google 工具栏功能。安装 Google 工具栏之后的 IE 浏览器将信息浏览和信息检索有机地结合起来,这种小小的功能改进确实是对用户的体贴和帮助。

在 Web 服务端,2000 年以后出现了几种主要的技术融合方式。首先,越来越多的 Web 开发环境开始支持 MVC(Model-View-Controller)的设计模型,为开发者提供全套的开发框架。实际上,J2EE 和.NET 平台本身就是这种开发框架的典型代表。其次,门户服务(Portal Server)和 Web 内容管理(Web Content Management)在最近几年里成为了应用集成的重点模型。这两种应用模型可以直接为开发者或最终用户提供构建 Web 应用的高级平台,可以让 Web 开发和信息发布工作大为简化。在商业软件领域,这一类应用的例子包括 Microsoft 的 SharePoint、IBM 的 WebSphere Portal、FileNet 的 Web Content Manager 等。开源项目在 Web 开发框架和应用模型方面表现得非常积极,Struts、Jetspeed、jPortlet、Cocoon、Lenya、XOOPS、Spring 等都是开源世界里与 MVC 开发框架、门户服务和 Web 内容管理相关的优秀解决方案。

1.7 可扩展标记语言——XML

如果说 HTML 语言给 Web 世界赋予了无限生机的话,那么,XML 语言的出现大概就可以算成是 Web 的一次新生了。按照伯纳斯·李的说法,Web 是一个“信息空间”。HTML 语言具有较强的表现力,但也存在结构过于灵活、语法不规范的弱点。当信息以 HTML 语言的面貌出现时,Web 这个信息空间是杂乱无章、没有秩序的。为了让 Web 世界里的所有信息都有章可循、有法可依,需要一种更为规范、更能够体现信息特点的语言。所以,出现了 XML(eXtensible Markup Language,可扩展标记语言)。

1996 年,W3C 在 SGML 语言的基础上,提出了 XML 语言草案。1998 年,W3C 正式发布了 XML 1.0 标准。XML 语言对信息的格式和表达方法做了最大程度的规范,应用软件可以按照统一的方式处理所有 XML 信息。这样一来,信息在整个 Web 世界里的共享和交换就有了技术上的保障。HTML 语言关心的是信息的表现形式,而 XML 语言关心的是信息本身的格式和数据内容。从这个意义上说,XML 语言不但可以将客户端的信息展现技术提高到一个新的层次,而且可以显著提高服务端的信息获取、生成、发布和共享能力。为了将 XML 信息转换为 HTML 等不同的信息展现形式,1999 年,W3C 制定了 XSLT(eXtensible Stylesheet Language Transformation,扩展样式表转换语言)标准,XSLT 是专门用来转换 XML 文档结构的语言。同一年,IE 5.0 增加了对 XML 和 XSLT

的支持。现在，网站的开发者可以直接使用 XML 语言发布信息。

1.8 Web 开发技术的未来

未来 Web 会是什么样子呢？W3C 是这样回答的，未来的 Web 是语义化的 Web (Semantic Web)。今天的 Web 可以自如地生成、传递和展现各式各样的信息，但它还只是一个信息的“容器”，很难揭示出信息本身的内容和特性。未来的语义化 Web 是一种懂得信息内容的 Web，是真正的“信息管理员”。

从技术角度看，XML 语言统一了信息的表达方式，还不能揭示信息的内容。1998 年，W3C 和一些研究机构开始对元数据（Metadata）进行研究。元数据是描述数据的数据，可以揭示信息的内容特性。1999 年，NetScape 提出的 RSS（Rich Site Summary）建议标准是用元数据技术描述新闻等信息内容的第一次尝试。1999 年，W3C 的研究小组提出了 RDF（Resource Description Framework）标准草案。RDF 在 XML 语法的基础上，规定了元数据的存储结构和相关的技术标准。使用 RDF 语言，可以用统一的、可交换的格式揭示出信息本身的各种特性。2001 年，W3C 又开始着手制定 OWL（OWL Web Ontology Language）标准。OWL 语言也是一种符合 XML 标准的语言，它比 RDF 又前进了一步，可以更加深入、细致地描述信息内容。在 RDF 和 OWL 语言的帮助下，能让 Web 上的信息内容变得更容易理解、更便于交换和共享。2003 年，W3C 成立了语义化 Web Service 研究小组（Semantic Web Services Interest Group），研究在 Web Service 中加入语义技术的相关问题。2004 年 2 月，W3C 宣布 RDF 和 OWL 标准正式成为 W3C 的建议方案，这标志着语义化 Web 的大厦已经破土动工。

随着语义化 Web 的诞生和发展，Web 开发技术也必将经历更为重大的变革。可以预见的是，在未来的时间里，还会有许多新的开发技术或开发平台出现。从静态技术到动态技术，从开发平台到应用模型，从传统 Web 到语义化 Web，等等，为了让更多的人获得更有价值的信息服务，Web 开发者们可能还会经历一次又一次的技术浪潮，将会面临更为严峻的技术挑战与更多的机遇。

1.9 学习的准备

本节主要介绍本书将带领读者学什么，学习目标是什么，学习中应该准备什么工具。

1.9.1 学什么

以上介绍了 Web 的基本概念，概要介绍了 Web 开发领域中使用的一些典型技术，通过上面各节的内容可以发现 Web 虽然历史不长，但其开发技术类型却很多。如何在有限的时间内掌握最基本的 Web 开发技术呢？这就是本书要解决的问题，本书将带领读者使用 HTML、JavaScript、CSS、XML、JSP 几种基本的 Web 开发技术，将理论知识与上机实践相结合，学习目标是融会贯通所学的基本 Web 开发技术开发出一个具体的 Web 网站，

为今后学习其他 Web 开发技术打下基础。

1.9.2 Web 开发工具

要学习 HTML、JavaScript、CSS、XML、JSP 语言，意味着要学习编写不同类型的 Web 页文件。要开发一个 Web 网站，意味着要开发一系列相关的 Web 页文件。因此，在学习 Web 开发技术之前，需要了解并选择编写 Web 页的工具，选择好学习工具可以起到事半功倍的作用。

编写 Web 页最简单的工具是记事本。不过，现在市面上已经有很多 Web 页编辑器，它们使编制 Web 页的工作变得既简单又容易，例如 FrontPage、Dreamweaver 等。Word 中也有编制 Web 页的功能，如果用它编辑页面也是很方便的。下面介绍几种典型的 Web 开发工具。

1. FrontPage

FrontPage 是优秀的 Web 开发编辑器，可以方便、快捷地编辑 Web 页。在编辑器中提供了所见即所得、手工编辑 HTML 和预览 Web 页三种编辑模式。

它是一个非常简单实用的网页编辑软件，用它可以非常快速地编辑好一个网页，特别适用于第一次做网页的人。FrontPage 还可以做简单的多媒体课件、企业策划书、产品演示说明书等，它具有上手快，易学、易用的特点，是初学者理想的制作网页的集成工具。

2. Dreamweaver

Dreamweaver 是美国 Macromedia 公司开发的集 Web 页制作和管理网站于一体的所见即所得 Web 页编辑器，它是第一套针对专业 Web 页设计师的视觉化 Web 页开发工具，利用它可以轻而易举地制作出跨越平台和跨越浏览器的充满动感的 Web 页。是目前优秀的制作网页的集成工具之一。

3. TopPage

IBM 推出的 NetObjects TopPage 是一种既适合专业网络开发人员又适合初学者的工具。NetObjects TopPage 是一种全能的工具，小型企业、个人和学校均可用它很方便地建立外观优美的网站。此外，TopPage 还是一种先进的开发环境，在此环境中，网络设计管理者可用动态 HTML 进行链接或制图、编辑级联式表格、建立可编辑的站点分布图、修剪或滚动图像、建立 JavaScript 应用程序、创立和管理框架。总之，TopPage 可以创建在所有浏览器上都能运行自如的网页，使用起来极其方便。

4. Eclipse

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，可以通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集，包括 Java 开发工具(Java Development Tools，JDT)，还包括插件开发环境(Plug-in Development Environment，PDE)，这个组件主要针对希望扩展 Eclipse 的软件开发人员，

因为它允许他们构建与 Eclipse 环境无缝集成的工具。

Eclipse 就像软件开发者的“打铁铺”,一开始只有火炉、铁钻与铁锤。但铁匠会用现有的工具打造出新的工具,使用 Eclipse 可以打造出进行不同软件开发的工具。例如,通过添加 Eclipse 匹配的汉化包插件可以汉化 Eclipse。通过添加 Visual Editor 插件可以建立编写 Java 类可视化的开发环境。通过添加 MyEclipse 与设置 Tomcat 服务器,可以建立开发 J2EE 应用的开发环境。

Eclipse 是一个开放源代码的项目,任何人都可以到 http://www.eclipse.org/downloads 下载 Eclipse 的源代码,并且在此基础上开发自己的功能插件,它是一款非常受欢迎的软件开发工具,也是进行 Web 开发常用的工具,实际上使用它的 Java 开发人员是最多的。

Eclipse 缺点就是较复杂,对初学者来说,理解起来比较困难。

1.10 思考与练习

1.10.1 思考题

1-1 说明 HTML 语言对网络的促进作用。

1-2 谈谈你对伯纳斯·李的看法。

1-3 HTML 语言和浏览器是什么关系?

1-4 本书中所指的静态和动态是什么意思? 通常所说的动态是什么意思?

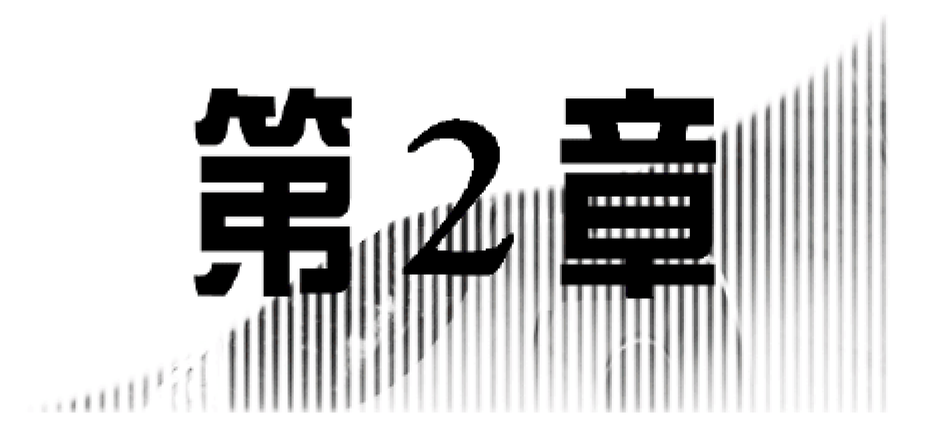
1-5 Web 客户端有哪些常用技术?

1-6 Web 服务端有哪些常用技术?

1-7 本章中出现了哪些英文缩写名称? 它们各代表什么含义?

1.10.2 上机练习

选择一种开发 Web 页的工具,上机了解其使用方法。



超文本标记语言——HTML

要开发网站,先要掌握编写网页的技术;要编写网页,先要了解 HTML 语言,即超文本标记语言,它是编写网页的基础。

HTML 语言由一些标记码、字母和文字组成,以普通的文本文件格式保存,可以用所熟悉的任何文字编辑器来编辑。通过 HTML 中简单的标记码,在浏览器中可以显示包含文本、图像、声音、视频等元素的声、图、文并茂的页面,通过超链接标记码可以实现当前页面到其他任何 Web 页面的超链接。

HTML 语言初期只有很少的一些标记,但由于在 HTML 页面上显示的内容越来越多,其标记在不断增加,通过 HTML 的最新版本可以了解最新最全的标记。本章主要介绍一些 HTML 的基本知识。

本章主要解决以下问题:

- 如何使用 HTML 语言编写 HTML 格式文件;
- 显示一个 HTML 页面使用哪些标记码;
- 显示不同文本格式使用哪些标记码;
- 在页面中实现超链接使用哪些标记码;
- 在页面中添加其他元素使用哪些标记码;
- 显示表单组件使用哪些标记码。

2.1 HTML 页面结构标记码

本节主要解决如何使用 HTML 标记码编写、保存 HTML 文件,如何在浏览器中浏览 HTML 文件、用哪些标记码定义 HTML 页面结构等问题。

2.1.1 开始与结束标记码

告诉浏览器开始显示一个 HTML 文档的标记码是<html>,它告诉浏览器下面的内容是 HTML 文档,在 HTML 文档结束处要有对应的</html>标记码,它告诉浏览器 HTML 文档结束了。

开始与结束标记码的语法格式:

```
<html>
  html 文件的所有内容
</html>
```

2.1.2　头部标记码

HTML 文档的头部标记码是<head>，它告诉浏览器下面的内容是 HTML 文档的头部，显示在文本之前，</head>标记码告诉浏览器头部内容结束了。

头部标记码的语法格式：

```
<head>
    HTML 文档的头部内容
</head>
```

头部内容一般包括注释、META 和标题。

2.1.3　网页标题标记码

<title>和</title>标记码对告诉浏览器标记码对之间所包含的文字为这个 Web 页的标题，一般写在头部标记码之中。网页标题文字会显示在 Web 浏览器最上面的 title（标题）栏上。浏览器用户可以把标题加入收藏夹中，所以网页标题文字最好使用中文并且有明确的意义。

网页标题标记码的语法格式：

```
<title>Web 页的标题</title>
```

下面用一个例子来说明如何使用 HTML 语言编写一个 HTML 页面，虽然有点麻烦（使用网页编辑器可以自动生成一个 HTML 页），但能帮助你弄清 HTML 文档的结构。

例 2.1　编写一个具有头部内容和网页标题的 HTML 文档（2-1.html 文件）。

编写 HTML 文档的步骤如下：

（1）打开记事本，输入以下标记码和文本文字

```
<html>
<head>
<title> 2-1 这里显示的文字是：Web 页标题</title>
这里显示的是：Web 页面头部标记码中的文字
</head>
</html>
```

（2）将输入的内容保存为名称为 2-1.html 的文本类型（.txt）文件，注意其保存路径，例如 E:HTML。

（3）在 IE 浏览器菜单栏单击“文件”→“打开”菜单，在 E 盘 HTML 文件夹中选择 2-1.html，即可通过 IE 浏览器显示你编写的 HTML 网页了，结果如图 2.1 所示。

2.1.4　主体标记码

<body>标记码是 HTML 文档主体标记码的开始，结束符为</body>。HTML



图 2.1 在浏览器中显示的 HTML 页面

页面的主要内容都出现在＜body＞和＜/body＞标记码对之中。HTML 文档中只能有一对 body 标记码，且它只能跟在 head 标记码后面。

主体标记码的语法格式：

```
<body>
    主体内容
</body>
```

例 2.2 一个使用主体标记码在 HTML 文档(2-2.html 文件)。

```
<html>
<head>
<title>2-2</title>
这里显示的是：Web 页面头部标记码中的文字
</head>
<body>
这是使用 HTML 语言编写的 Web 页面文件
</body>
</html>
```

2-2.html 在 IE 浏览器里的显示结果如图 2.2 所示。

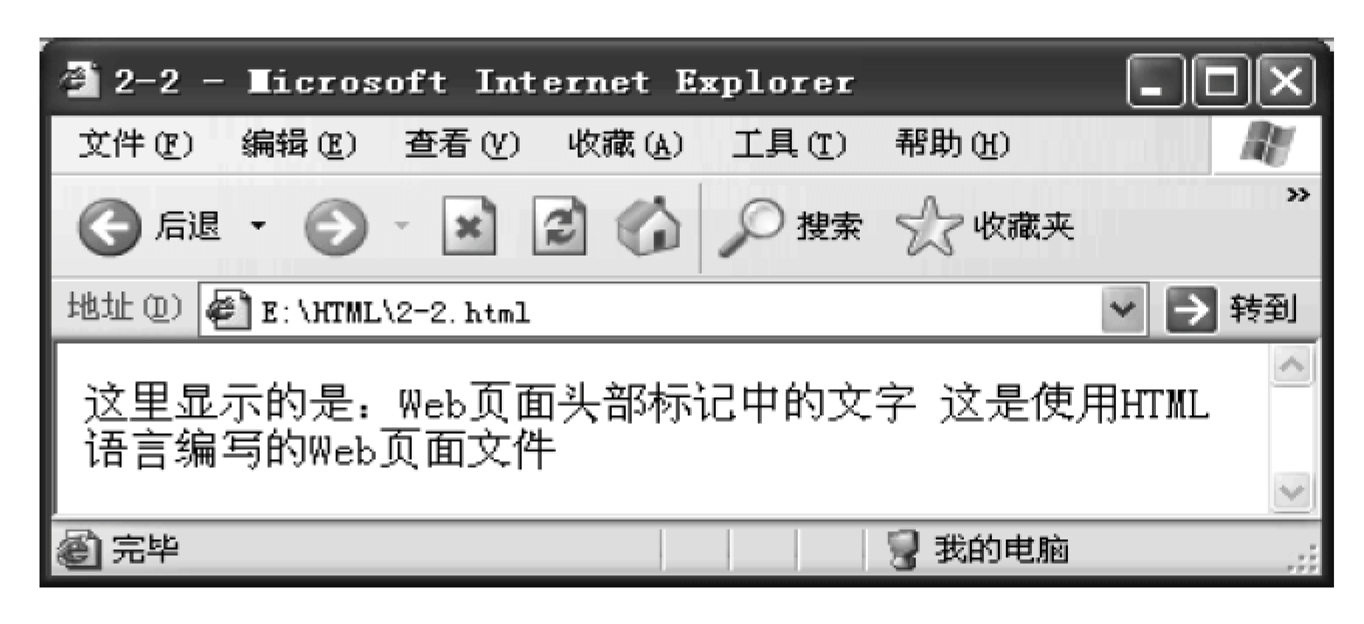


图 2.2 添加主体内容的页面

注意：从以上四对标记码可以看到 HTML 语言的标记码是由英文字母与“＜”、“＞”、“＜/”符号构成的，开始为＜标记码＞，结束为＜/标记码＞，这一点在编写 HTML 语言时一定要牢记。

HTML 语言的标记码是成对出现的，不过，也有个别的标记码是单独出现的，以后会看到。

标记码使用的英文字母一般为小写，其实大小写都可以，浏览器不区分字母的大小写。

2.1.5 通过设置主体标记码的属性定义页面背景

<body>标记码还具有一些属性，例如，背景色、链接的颜色、文字的颜色等。

带有属性的主体标记码语法格式：

```
<body [bgcolor=# |text=# |link=# |alink=# |vlink=# |background=图像文件名]>
    文档内容
</body>
```

其中，bgcolor 为页面背景颜色属性，text 为页面中文字的颜色属性，link 为页面中待连接超链接对象的颜色属性，alink 为页面连接中超链接对象的颜色属性，vlink 为页面中已连接超链接对象的颜色属性，background 指定 HTML 背景为平铺的图像。

各属性的参数值可以是用英文 red、blue、yellow 等，也可以是＃与 6 位十六进制数，每种颜色用两位十六进制数，从 00 到 FF，前两位设置红色的深浅，中间两位设置绿色的深浅，后两位设置蓝色的深浅。例如红色为＃ff0000，蓝色为＃0000ff 等。各属性可选也可不选。

注意：方括号[]中的属性可选可不选，竖线(|)表示可同时选的属性。根据需要可以选择其中的一个、部分或全部。

例 2.3 添加文字颜色和背景颜色属性的文档(2-3.html 文件)。

```
<html>
<head>
<title>2-3</title>
这里显示的是：Web 页面头部标记码中的文字
</head>
<body bgcolor="yellow" text="#000000">
这是使用 HTML 语言编写的 Web 页面文件
</body>
</html>
```

文件 2-3.html 在 Web 浏览器的显示结果如图 2.3 所示。不过，颜色从本书上是看不出来的，可以亲手试一试。



图 2.3 添加颜色属性的页面

注意：在编写 HTML 文件时要采取“编写—存盘—浏览”三步曲，以此循环工作。

2.1.6　说明信息标记码

在网页的头部还可以看到 meta 标记码，它用来描述说明信息，包括文档的语言字符集信息、作者信息、文档过期时间、关键词列表、页面的自动刷新属性等。meta 标记码的内容在浏览器中不显示。

说明信息标记码的语法格式：

```
<meta http-equiv="特征名" content="特征值[; charset=#]">
```

例如：

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta http-equiv="作者" content="张驰">
<meta http-equiv="文档期限" content="2008/9/25">
<meta http-equiv="refresh" content="10">
```

refresh 用来设置页面的自动刷新属性，时间为 10 秒。

注意：meta 一般放在头部内。常用的语言字符集有 x-mac-roman、gb2312、gb_2312-80、big5 等。

2.1.7　注释标记码“＜!”与“＞”

为了使程序清楚、容易理解，可以添加注释标记码“＜!”和“＞”(也可写为“＜!--”和“--＞”)，在注释标记码这对符号之间的内容不会被浏览器显示出来。

＜!DOCTYPE…＞用来在 HTML 文档的第一行显示版本信息。例如：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

2.2　定义文本格式的标记码

在例 2.3 中可以看到头部文字和主体文字是连在一起的，而且字体大小一样，不能分辨哪里是头部文字，哪里是主体中的文字。HTML 语言是如何解决这个问题的呢？

本节的内容主要解决用哪些标记码在 HTML 文档中显示不同格式的文本文字。

2.2.1　文本标题标记码

HTML 通过文本标题标记码来显示作为标题的文本文字。通过文本标题标记码告诉浏览器将这些文本文字以指定标题(黑体字)格式显示在指定位置。

文本标题标记码为＜hi＞＜/hi＞。被＜h1＞和＜/h1＞夹在中间的文本文字，为 1 号标题。依次可以标注出 6 个层级的子标题，从＜h1＞到＜h6＞。差别在于标题数字小的文本会比标题数字大的文本大些、粗些、更显眼。在下面的例子里可以看见它们之间的差别。每遇到一个文本标题标记码时，当前段落就会被终止，作为标题的文本前后会各留

出一定的空白。标题字自动从下一行开始。

文本标题标记码的语法格式：

```
<hi>标题文字</hi>  (i=1,2,…,6)
```

例 2.4 使用不同文本标题标记码的文档(2-4.html 文件)。

```
<html>
<body>
<h1>这是 h1 标题字体!!</h1>
<h3>这是 h3 标题字体!!</h3>
<h5>这是 h5 标题字体!!</h5>
</body>
</html >
```

文件 2-4.html 在浏览器里显示的结果如图 2.4 所示。

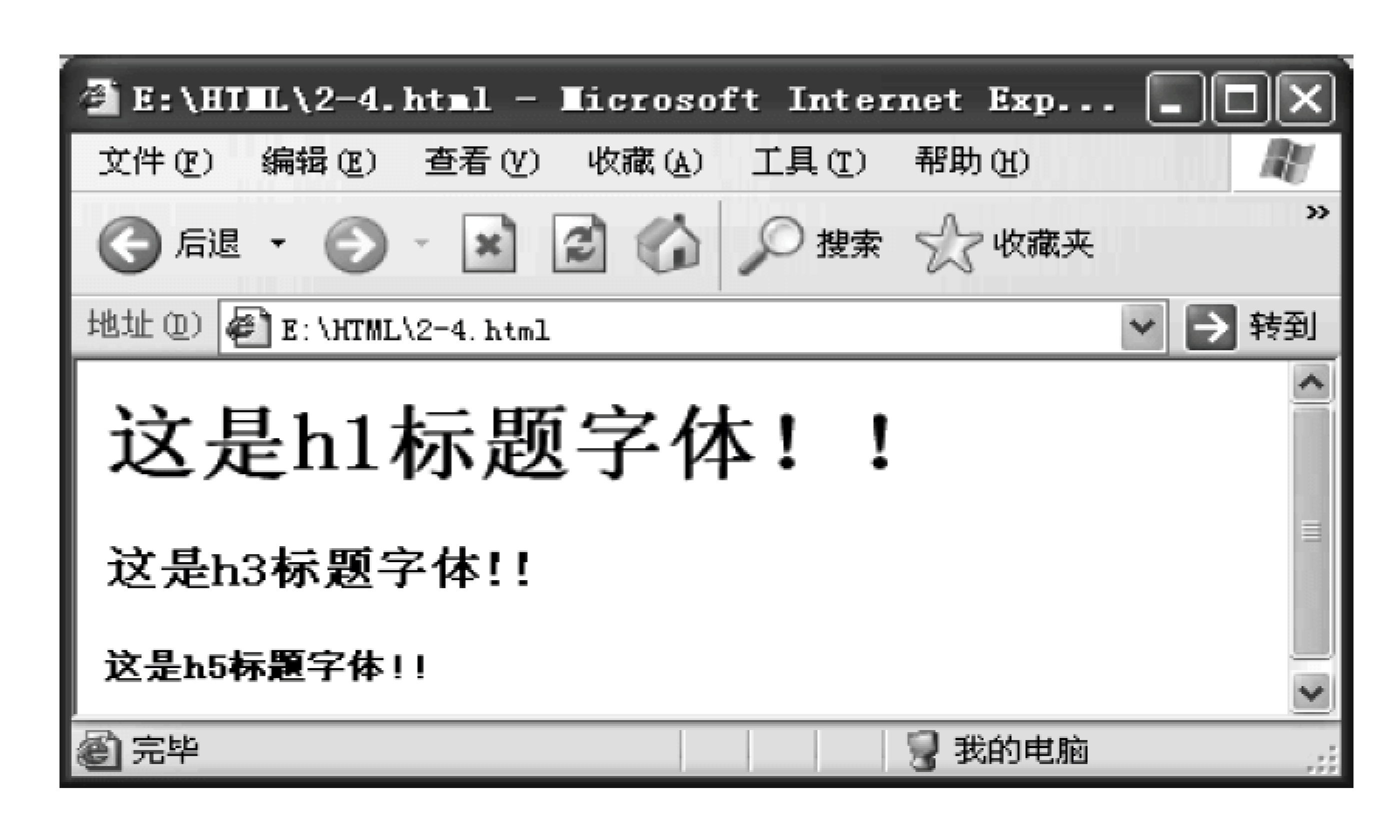


图 2.4　显示不同标题文字的页面

2.2.2　字体标记码及大小和颜色属性

通过字体标记码及其属性能够改变包含的字符、单词、短语或指定范围文本的字体大小和颜色。

字体标记码的语法格式：

```
<font size=# color=#>文字</font>
```

Size 属性用来设置文本字体的大小(属性值为 1～7,或者相近的数字),默认为 3 号字体。Color 属性与主体中的 color 属性相同。

例 2.5 设置文字字体大小和颜色的文档(2-5.html 文件)。

```
<html>
<body>
<font size=7 color=#ff0000>这是 7 号字体!!</font>
<font size=5 color=#0000ff>这是 5 号字体!!</font>
<font size=3>这是 3 号字体!!</font>
```

```
<font size=1>这是 1 号字体!!</font>
</body>
</html >
```

文件 2-5.html 在浏览器显示的结果如图 2.5 所示。

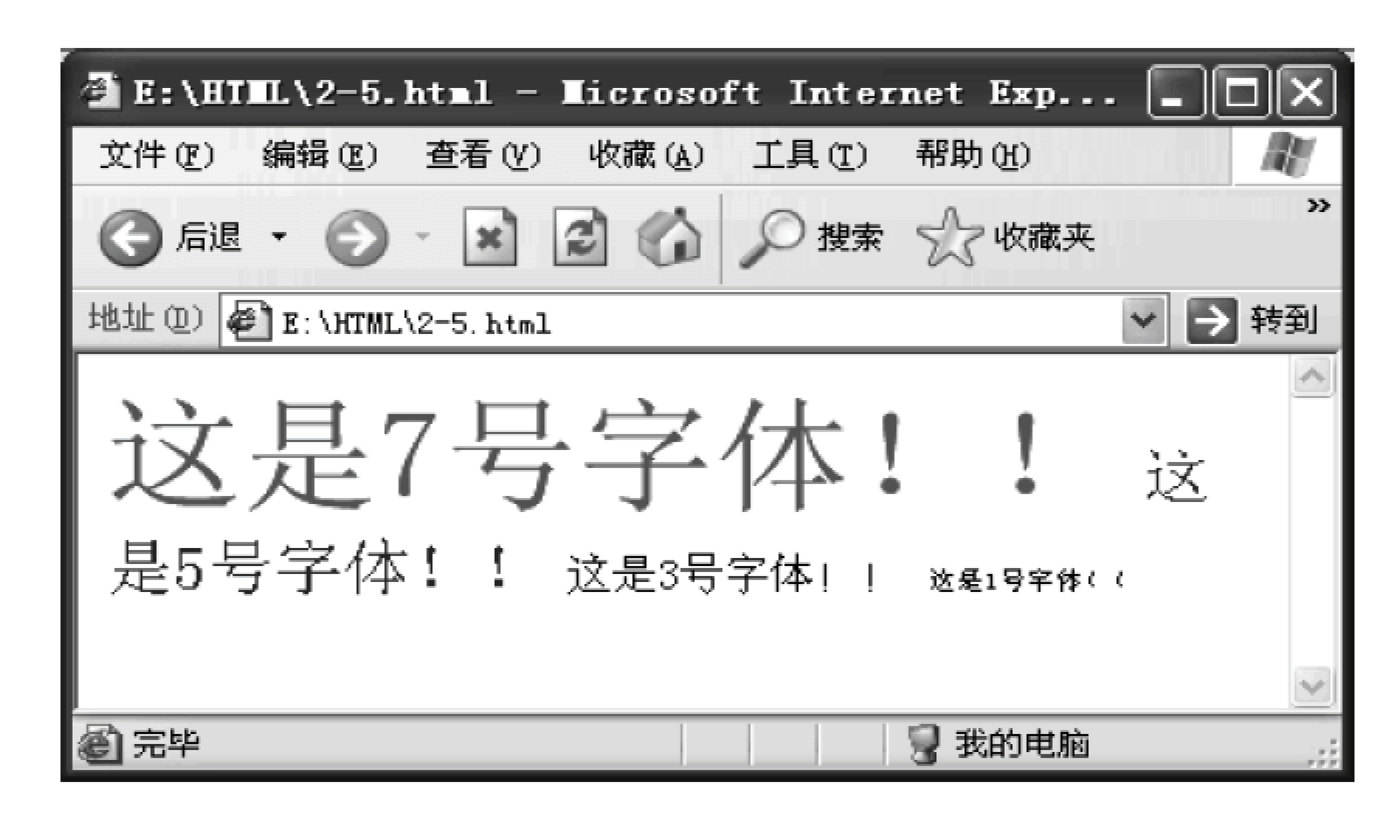


图 2.5　显示不同字体文字的页面

注意：字体标记码不会自动换行，这一点与标题字体不同。字体标记码一般在大块的正文文字中使用。

2.2.3　段落标记码与换行标记码

在 HTML 文档里不需要在意文档每行的宽度，不必担心文字是不是太长了而被截掉，它会根据页面窗口的宽度自动换行。

<P>是 HTML 格式中特有的段落标记码。<P>标记码格式浏览器页面中的文本文字在这里分为一段，下面的文字要成为一个新的段落从头再开始。如果不使用<P>标记码，浏览器会把所有的文字都排在一个段落里，遇到窗口的边界自动换行。文本文件中的换行标记浏览器是不认识的。

是 HTML 格式中特有的换行标记码。
标记码告诉浏览器页面中的文本文字在这里要换行，下面的文字要换行从头再开始。

段落标记码与换行标记码的语法格式：

```
<P>文本</P>
文本<br>
```

下面的例子可以看出二者的区别。

例 2.6　使用段落和换行标记码的文档(2-6.html 文件)。

```
<html>
<body>
    在 HTML 文档里不需要在意文档每行的宽度,不必担心文字是不是太长了而被截掉,(加段落标
记码另起一段)<P>会根据页面窗口的宽度自动换行。</P>标记码告诉浏览器页面中的文本文字
在这里要换行 (加换行标记码),<br>下面的文字要换行从头再开始。
</body>
```

```
</html>
```

文件 2-6.html 在浏览器显示的结果如图 2.6 所示。

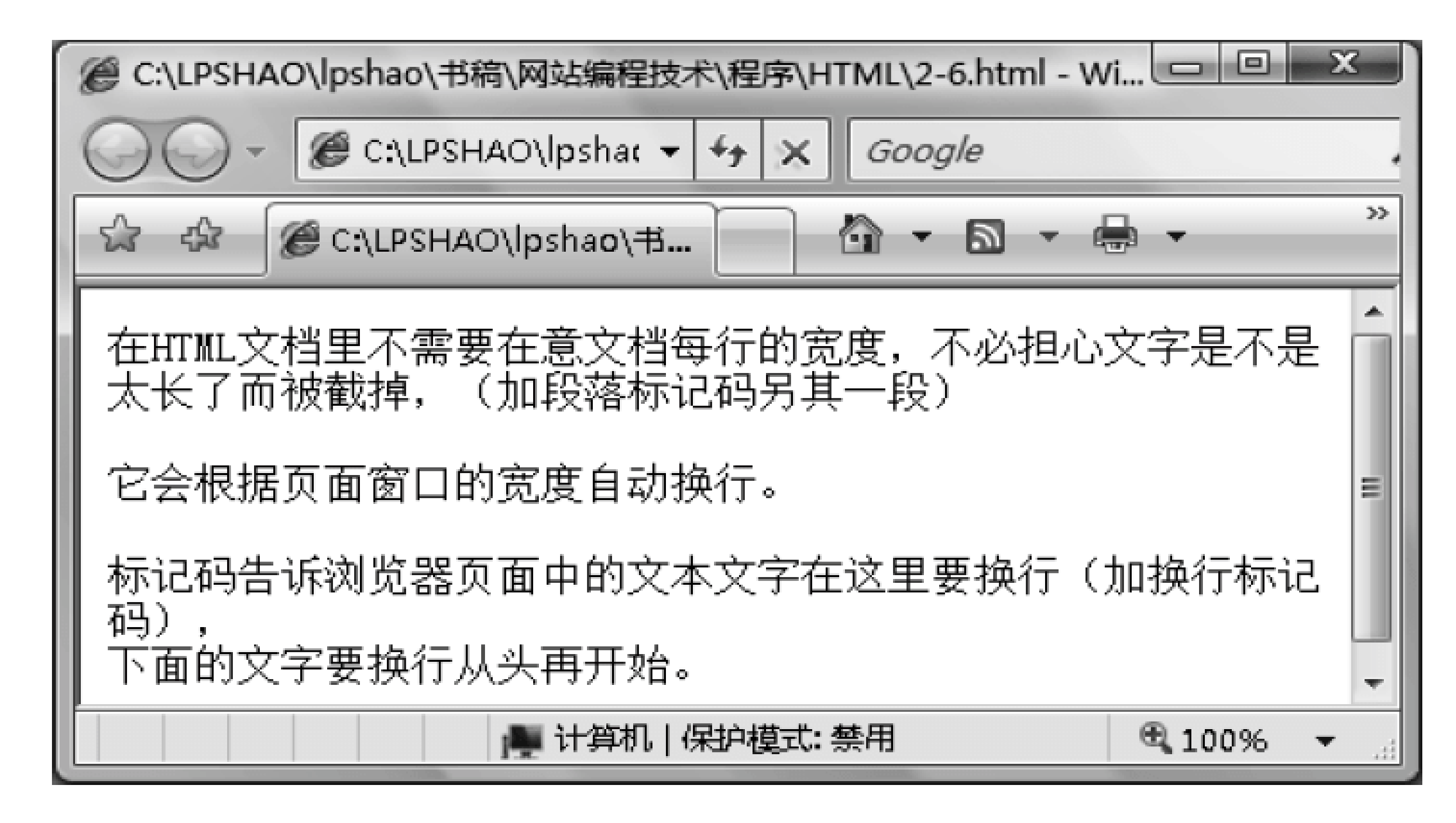


图 2.6　使用段落与换行标记码的页面

注意：
标记码是单独使用的。

2.2.4　字符格式标记码

对特殊的字或句子可以使用字符格式标记码在浏览器中以不同格式显示，常用的文本格式标记码有：斜体标记码<i>、强调标记码<em>、黑体标记码<b>、着重强调标记码<strong>，下划线标记码<U>，打字机字体标记码<tt>。还有：

- <big>　显示大型文本文字；
- <code>　标记 HTML 指令；
- <dfn>　标记定义的语句；
- <kbd>　标记键盘字符；
- <pre>　按照原来键入的方式原样显示文字间隔、换行、空白等；
- <S>　给文本加上删除线标记；
- <samp>　标记码某个命令的例子；
- <small>　显示小号尺寸的文本；
- <sub>　创建下标；
- <sup>　创建上标；
- <var>　标记程序变量。

注意：这些标记码都是成对使用的。

例 2.7　使用格式标记码显示文本的文档(2-7.html 文件)。

```
<html>
<body>
<i>斜体</i><em>(斜体)</em>
<b>黑体</b><strong>(黑体)</strong><br>
<tt>打字机字体</tt><var>程序变量 x y<sub>2</sub>  z<sup>3</sup></var><br>
<code>HTML 指令</code><samp>例 1</samp>
```

```
<kbd>使用键盘字符时 a%#@!()&*</kbd>
<big>用以显示大型文本</big>
<pre>
        看看是怎样使文本保持原有模样的!
#!/bin/csh  cd$SCR× * cfsgetmysrc.f:mycfsdir/mysrc.f
fc-02-omya.outmysrc.f  mya.out
</pre>
</body>
</html>
```

文件 2-7.html 在浏览器显示的结果如图 2.7 所示。



图 2.7 显示不同格式的字符页面

2.2.5 分区标记码

分区标记码的语法格式:

```
<div>一个分区的内容</div>
```

利用分区标记码<div> </div>可以将一个页面分成独立的几个部分,使分区中包含的文字具有自己的样式风格,而独立于文档的其他文字内容。利用其 class 属性可以为分区起一个名字。分区的属性还有 align。例如:

```
<div class=C1 align=center> …</div>
```

该分区的内容都将居中显示。

2.2.6 块引用标记码

块引用标记码的语法格式:

```
<blockquote>引用文字</blockquote>
```

<blockquote>与</blockquote>标记码之间一般包含较长的大块文本,浏览器在显示块中文本时会自动右移,左边空出几个格,将包含块中的文字与其他文字区别开。

2.2.7 地址标记码

<address>标记码通常用来告知本篇文件作者的联系地址、电话、E-mail 信箱。经常有一个 E-mail 地址放在文件的最后面。

地址标记码的语法格式：

```
<address>地址、电话、Email 信箱</address>
```

<address>…</address>标记码对定义文字为斜体。

2.2.8 几个特殊字符的写法

在 ASCII 码中，有三个字符：">"（大于）、"< "（小于）、"& "（和）被当作 HTML 文档中的控制码，要在屏幕上显示"<"字符需在 HTML 文档中输入"&lt"，显示">"字符需要输入"&gt"，显示"&"字符需要输入"&amp"。

还有两个常用符号分别代表著作权标志和商标符，可以在 HTML 文档中输入"&copy"与"&reg"，在浏览器中将显示©与®。

2.3 实现超链接的标记码

超链接标记码是 HTML 语言的重要标记，它使得网页具有了"超人"的功能，它可以在一个网页的不同段落间任意地链接，也可以从当前页面链接到另外一台 Web 服务器上的 Web 页面，还可以直接链接到本机的另一 Web 页面。

本节的内容主要解决用哪些标记码在 HTML 文档的不同页面之间、不同段落之间实现超链接功能，如何在 HTML 文档添加声音、视频信息等元素。

2.3.1 链接至本机另一 Web 页面

超链接标记码为<a>，链接到本机上的 Web 页面很简单，只要给<a>标记码的 href 属性设置文件名和路径就可以了。

超链接标记码的语法格式：

```
<a href="filename.html">链接到 filename</a>
```

在浏览器中会看到超链接标记码标记的"链接到 filename"超链接文字，其会变色并加上下划线，当鼠标移到上面时，鼠标箭头会变成小手状，单击超链接文字，浏览器会链接到同一机器同一路径的 filename. html 文件上。例如：

```
<a href="2-7.html">这里是"2-7.html"Web 页</a>
```

在浏览器中单击"这里是"2-7. html"Web 页"将显示 2-7. html 文件。

2.3.2 链接到其他服务器的 Web 页

链接到其他服务器上的 Web 页操作也很简单，只要把目的地的 URL 地址作为 href

的属性值就可以了。例如，想链接到“首都在线”的主页，可以编写如下语句：

```
<a href="http://www.263.net">这里是首都在线的主页</a>
```

超链接标记码的语法格式：

```
<a href="URL">这里是某地方</a>
```

例 2.8 使用超链接标记码从当前页面链接到其他 Web 页的文档(2-8.html 文件)。

```
<html>
<body> 例如：单击这里可以链接到本机的
<a href="2-7.html">"2-7.html"Web 页</a>
<p>链接到其他服务器上的 Web 页操作也很简单，只要把目的地的 URL 地址作为 href 的属性值就
可以了。</p>
  例如，想连接到"首都在线"的主页，可以单击这里：
<A href="http://www.263.net">www.263.net</A>即可。
</body>
</html>
```

文件 2-8.html 的显示结果如图 2.8 所示。

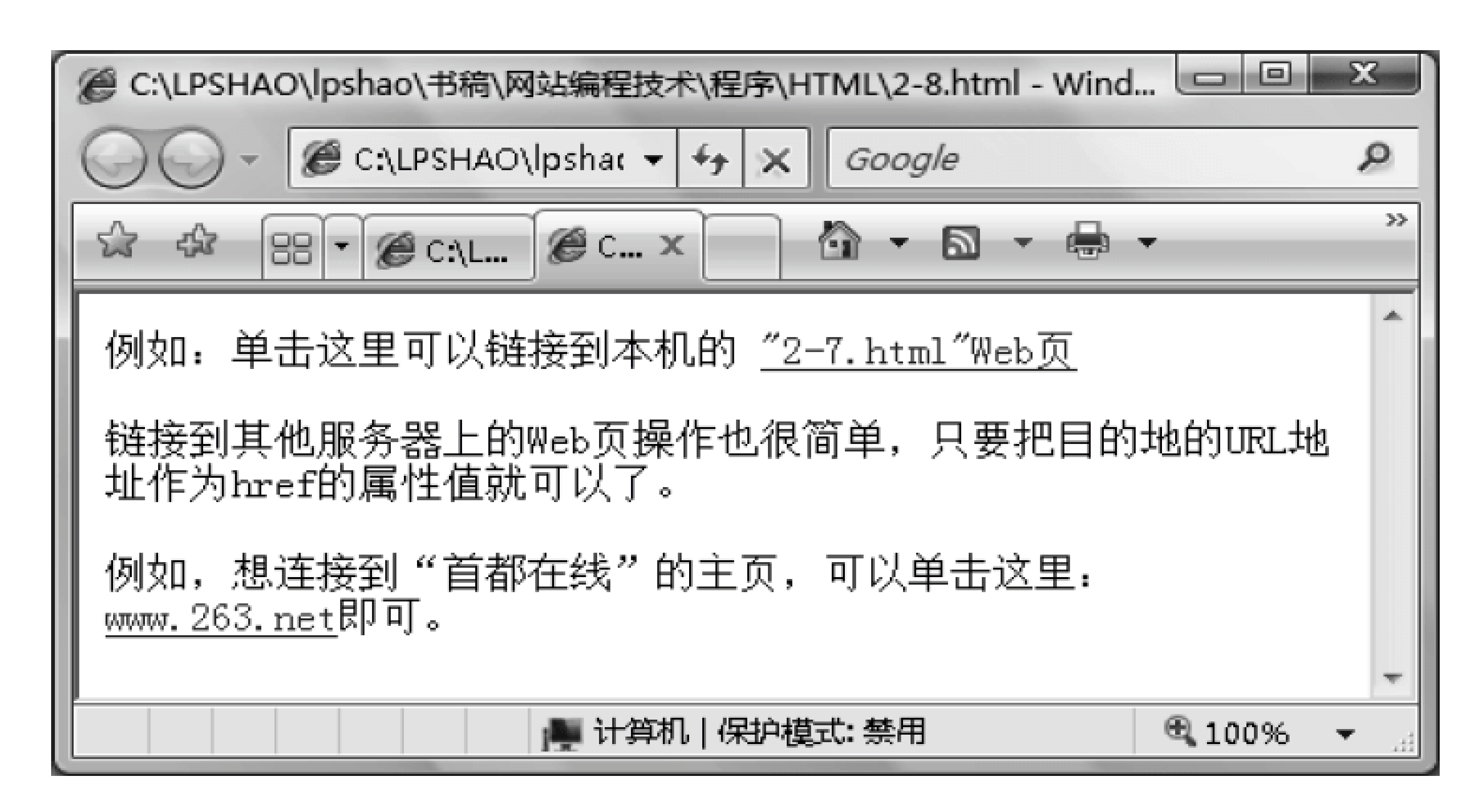


图 2.8 显示超链接的页面

注意：当链接到另一个 HTML 文件时，要做一个返回到本页的链接才可回到原来的页面，否则就不容易返回了。

2.3.3 链接到同一文件的另一个段落

通过超链接标记码不仅可以链接到其他 html 文件，也可以在同一个 Web 内进行链接。这和前面两种链接稍有不同，需要在链接的位置先做出一个“锚”标记，即链接的目标地，再做到“锚”的链接。例如，先在文件中选择“第一章”做“锚”标记码，第一章；再做到“锚”的链接，单击这里将显示第一章。

例 2.9 连接到同一 Web 页其他段落的文档(2-9.html 文件)。

```
<html>
```

```
<body>
<A name="第一章">第一章</A>为"锚"标记码。
<p>除了链接到另一个 HTML 文件,也可以在一篇文章内随心所欲地链接。<br>
这和前面两种稍有不同,需要先做出一个"锚"标记码,即链接的目标地,<br>
再做到"锚"的链接。到"锚"的链接与平常相同。</p>
<p></p><p></p><p></p><hr>
<A href="#第一章">单击这里将显示第一章</A>
</body>
</html>
```

文件 2-9.html 的显示结果如图 2.9 所示。

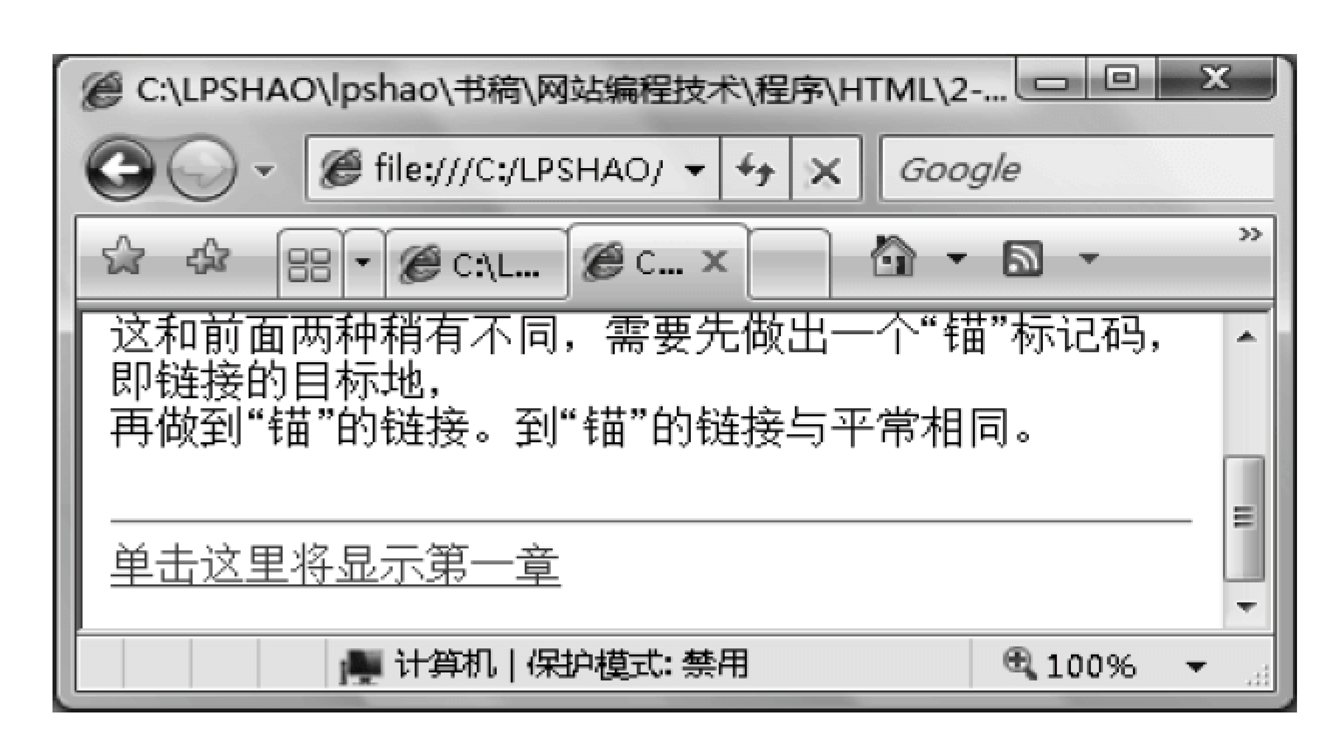


图 2.9 在同一文件中定义的超链接

注意：必须在不同的屏幕上才能看到链接效果，在同一屏幕看不出链接变化。如图 2.9 所示先移动到页面下方，再单击链接，即可看到链接到的文字"第一章"。

2.3.4 链接到不同文件的另一个段落

如果链接到不同文件的某个段落，也需要先设置"锚"标记。在 href 属性值中文件名后再添加上锚标记即可，例如：

```
<A href="2-9.html#第一章">到第一章</A>
```

在其他网页单击"到第一章"超链接文字就会显示 2-9.html 页面包含"第一章"段落的页面。

注意：除了用文字作为链接标记以外，还可以用图像。

2.3.5 链接音频和视频文件

在 Web 页面中加入音频和视频文件会使页面变得更加生动、活泼。在页面中加入音频和视频文件的方法非常简单，只要在超链接标记码的 href 属性中指定的 URL 网址、加上相应的音频和视频文件名称就可以了。例如：

```
<a href="yy.mid">在这里欣赏音乐</a>
<a href="test.mpeg">在这里欣赏视频</a>
```

无论音频信息还是视频信息都要求客户端有相应的应用程序可以播放这些信息。这

些程序或嵌入浏览器或独立存在。现在有的浏览器可以直接实现音频和视频信息的播放而不需要更多的附加软件。

注意：音频和视频的数据量较大，不可乱用。

2.3.6 链接到邮件编辑器

通过超链接的 href 属性，通过 mailto 值可以直接打开一个具有收件人地址的邮件编辑器，例如

```
<A href=" mailto:ZHC@163.NET">ZHC@163.NET</A>
```

2.4 显示其他元素的标记码

如果 Web 页面上全是文字，给人的感觉会很单调，怎样才能使页面活泼一些呢？怎样使页面显示出文本文字以外的其他信息元素呢？

本节的内容主要解决用哪些标记码在 HTML 文档显示图像、对象、表格等其他元素。

2.4.1 图像标记码

在页面上插入图像可使用 img 标记码。

图像标记码的语法格式：

```
<img src=图像的 URL [align=# alt=文字 border=# height=# width=#]>
```

img 标记码告诉浏览器这里要显示一个图像，src 属性告诉浏览器图像文件的来源即 URL 网址。

img 的属性有：align（显示的位置）、alt（显示文字）和 border（边框）、height（高度）、width（宽度）。

当浏览器关闭显示图像的功能时，通过 alt 属性显示 Text（文字）代替屏幕上一片空白或碎图片。Align 可设置图像显示的位置为 top（顶）、Middle（中间）、bottom（底）、Left（左）和 Right（右）。默认图像位置为文字的右边。如果图像很大，几乎充满整个窗口，可适当缩小。Border 属性可给图像加一个边框，其值取像素值，值为 0 时无边框。height（高度）和 width（宽度）属性用来确定在浏览器中显示的图像的看到与高度，值为像素值。

例 2.10 在页面中显示图像并链接一个音频文件的文档（2-10.html 文件）。

```
<html>
<body>插入一个图像
<img src=动物.gif  alt=动物>
<P><P><P>
加框并变小的图像
<img src=花.gif align=left border=2 height=100 width=200 alt=鲜花>
<br><P><P><a href="Water.gif">单击这里可以看到一个图像</a><P>
```

```
<P><P><P><A href="Sound.wav">单击这里可以欣赏音乐</A></body>
</html>
```

2-10.html 的显示结果如图 2.10 所示。

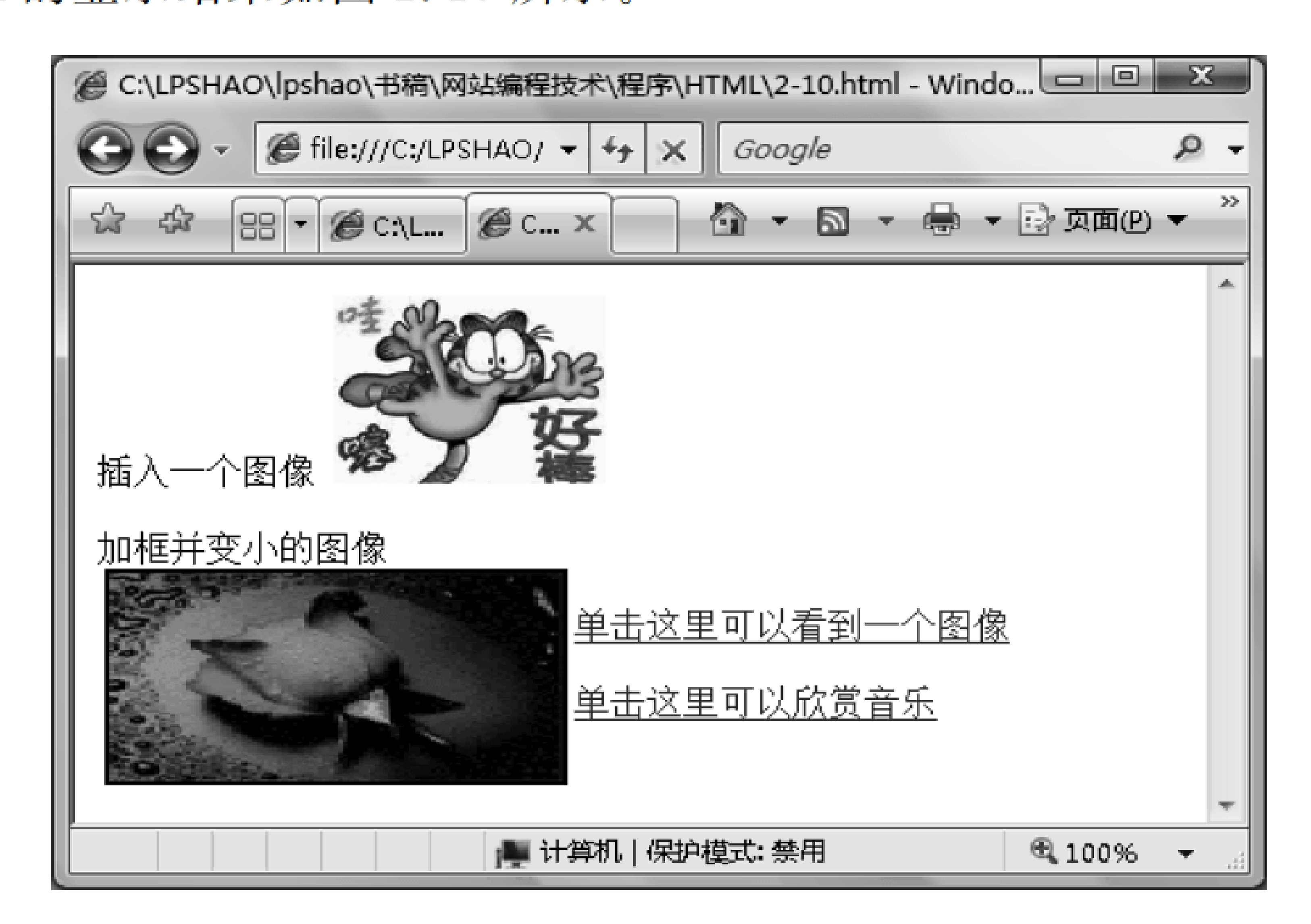


图 2.10　显示图片并链接声音文件的页面

注意：图像标记码<img>没有结束标记码。图像还可以作为链接的对象。不过，要用小图片，否则太浪费空间。当图片很大，或是希望读者选择性地看大图时，可以做一个链接，用几个文字或一个小图作为超链接，单击时打开大图窗口(另一个文件)，图 2.10 中就有这种情况。

2.4.2　对象标记码

在 Web 页还可以使用标记码 OBJECT 插入一些已经定义好的对象。

对象标记码的语法格式：

```
<object>…</object>
```

例 2.11　显示一个插入的日历控件对象的文档(2-11.html 文件)。

```
<html>
<body><center><h3>插入一个对象</h3></center>
<object id="日历控件 10.0" width="300" height="200" classid="CLSID:8E27C92B-
1264-101C-8A2F-040224009C02">
</object>
</body>
</html>
```

2-11.html 文件显示结果如图 2.11 所示。

使用<applet>…</applet>标记码可以插入一个使用 Java 语言编写的 applet 应用程序。



图 2.11 在页面中显示的日历对象

2.4.3 表格标记码

加入表格可使页面整齐直观，但写起来非常烦琐，这里只介绍它的部分语法格式，最好使用编辑器来编写表格标记码。

表格标记码的语法格式分为如下 5 部分。

1. 表格标记码

```
<table>…</table>
```

<table>表示下面是一个表格的内容的开始，</table>表示表格结束。

2. 表格标题标记码

```
<caption align=#>表格标题</caption>
```

表示一个表格的标题，也可不要。align 可选择 top(放在表格上面居中)、bottom(放在表格下面居中)，默认时标题放在表格上面居中。

3. 表格行标记码

```
<tr>…</tr>
```

表示表格一行的开始和结束。

4. 列名标记码

```
<th>列名</th>
```

在<th></th>中间加入列(字段)名，有几个列名就加入几个<th></th>标记码。

5. 列数据标记码

```
<td>数据</td>
```

在<td></td>中间加入该列数据，一般有要加入(行×列)个列标记码 <td></td>。

表格标记码具有如下属性：

- border=#表示表的边框，取像素值，默认时表格没有边框；
- cellspacing=#表示格边宽，取像素值，默认值为 2；
- cellpadding=#表示边框和内容间的距离，取像素值，默认值为 1；
- colspan=#表示占用列数，默认值为 1；
- rowspan=#表示占用行数，默认值为 1；
- width=#%表示表格宽度。

例 2.12 显示两个表格(一个不带框与一个带框)的文档(2-12.html 文件)。

```
<html>
<body><table>
<caption>价格表</caption>
<tr><th>日期</th><th>香菇</th><th>青椒</th></th>
<tr><td>9.2</td><td>28.00 元</td><td>0.90 元</td></th>
<tr><td>9.3</td><td>25.00 元</td><td>0.93 元</td></tr>
</table><hr>
<table BORDER=4>
<caption>价格表</caption>
<TR><TH>日期</TH><TH>香菇</th><th>青椒</th></tr>
<tr><td>9.2</td><td>28.00 元</td><td>0.90 元</td></tr>
<tr><td></td><td>25.00 元</td><td>0.93 元</td></tr>
</table>
</body>
</html>
```

2-12.html 显示结果如图 2.12 所示。

注意： 如果想加入一个空白格，在<TH ></TH>或<TD></TD>之间不加内容即可。表格的属性有很多，可以变宽度，可以建立链接，可以改变颜色等，你可以自己试一试。如果要使用表格，最好用编辑器的工具直接插入。在网页中使用太大的表格，会降低浏览速度。

2.4.4 地图标记码

使用地图标记码可以让浏览页面的用户通过地图链接到指定的地点，用户在地图图像的任意一处单击时，浏览器将会输出与这个图像热区超链接的 Web 页。

地图标记码的语法格式：

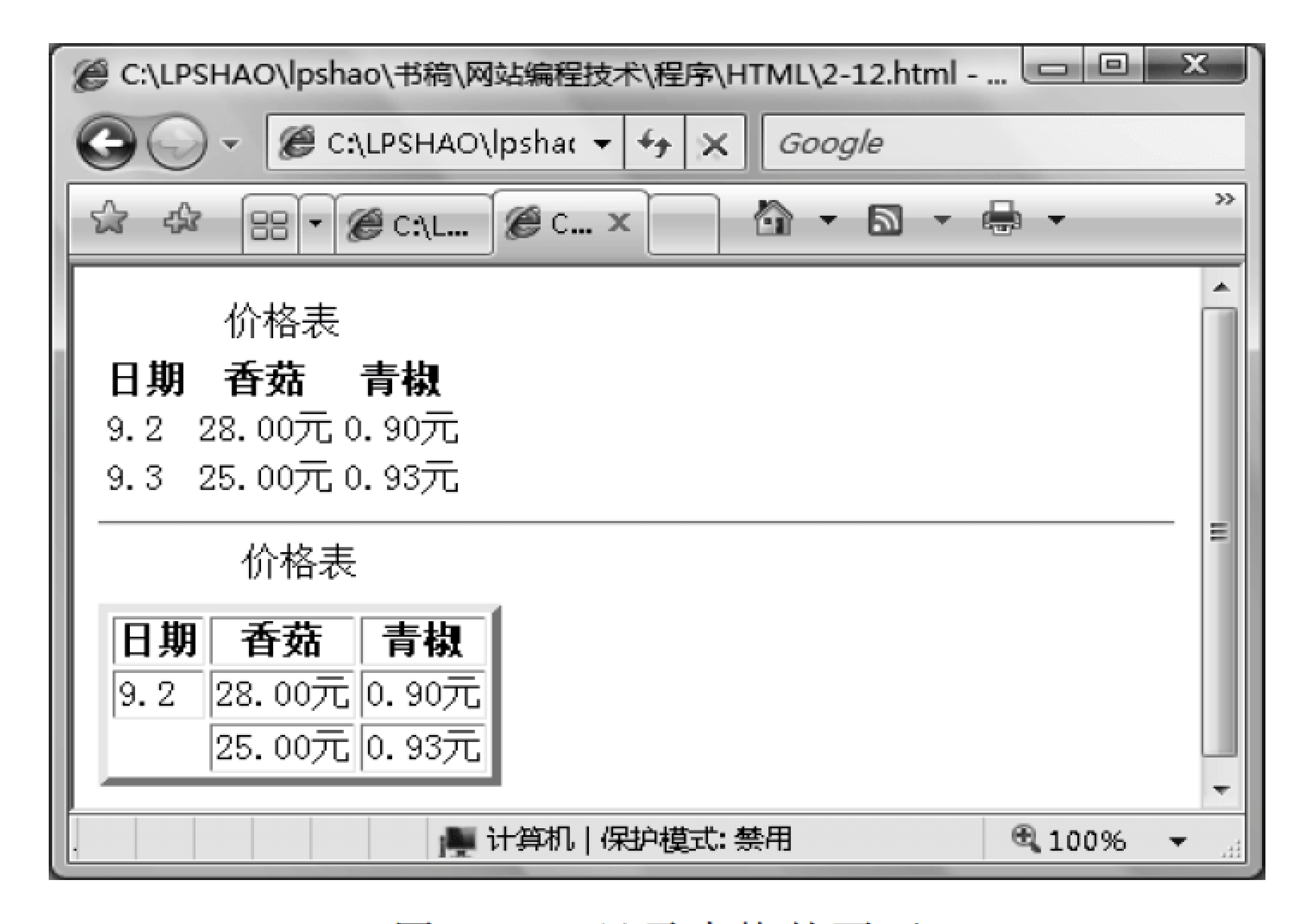


图 2.12　显示表格的页面

```
<img src=img.gif usemap="map-name">
<map name="map-name ">
<area shape="# " coords="# " href="url">
</map>
```

通过<img>标记码的属性 usemap 定义作为地图文件的名字与显示的地方，usemap="#"表示图像文件包含在 HTML 文档内，不带#表示图像为一个独立文件。<img>标记码的属性 src 指定作为地图的图像来源。

<map>标记码定义一个地图文件，必须带有 name 属性，name 属性的值是一个地图文件的唯一名称。

<area>标记码描述图像中一个热区的位置，可有任意个<area>标记码项。

属性 shape="rect"，表示热区形状为矩形；属性 coords="A，A′，B，B′"表示矩形坐标，(A，A′)为左上角坐标，(B，B′)为右下角坐标；

shape="circle"，表示热区形状为圆形，coords="A，A′，R′"表示圆坐标，(A，A′)为圆心坐标，R 表示圆半径；

shape="poly"，表示热区形状为多边形，coords="A，A′，B，B′，C，C′…"表示多边形坐标，(A，A′)为第一个角坐标，(B，B′)为第二个角坐标，……第 n 个角坐标；

shape="default"表示所有非热区的区域。

例 2.13　一个带有地图标记码的文档(2-13.html 文件)。

```
<html>
<body><img src="山.GIF" usemap="#Face">
<map name="Face">
<area shape="rect" href="2-7.html" coords="10,16,127,116">
<area shape="rect" href="2-8.html" coords="213,16,383,132">
<area shape="poly" href="2-9.html "coords="10,163,84,163,190,304,2,306">
<area shape="circle" href="2-10.html" coords="186,151,110">
```

```
</map>
</body>
</html>
```

在图 2.13 所示图像中用鼠标到处操作一下，就会理解“地图”的意思了。

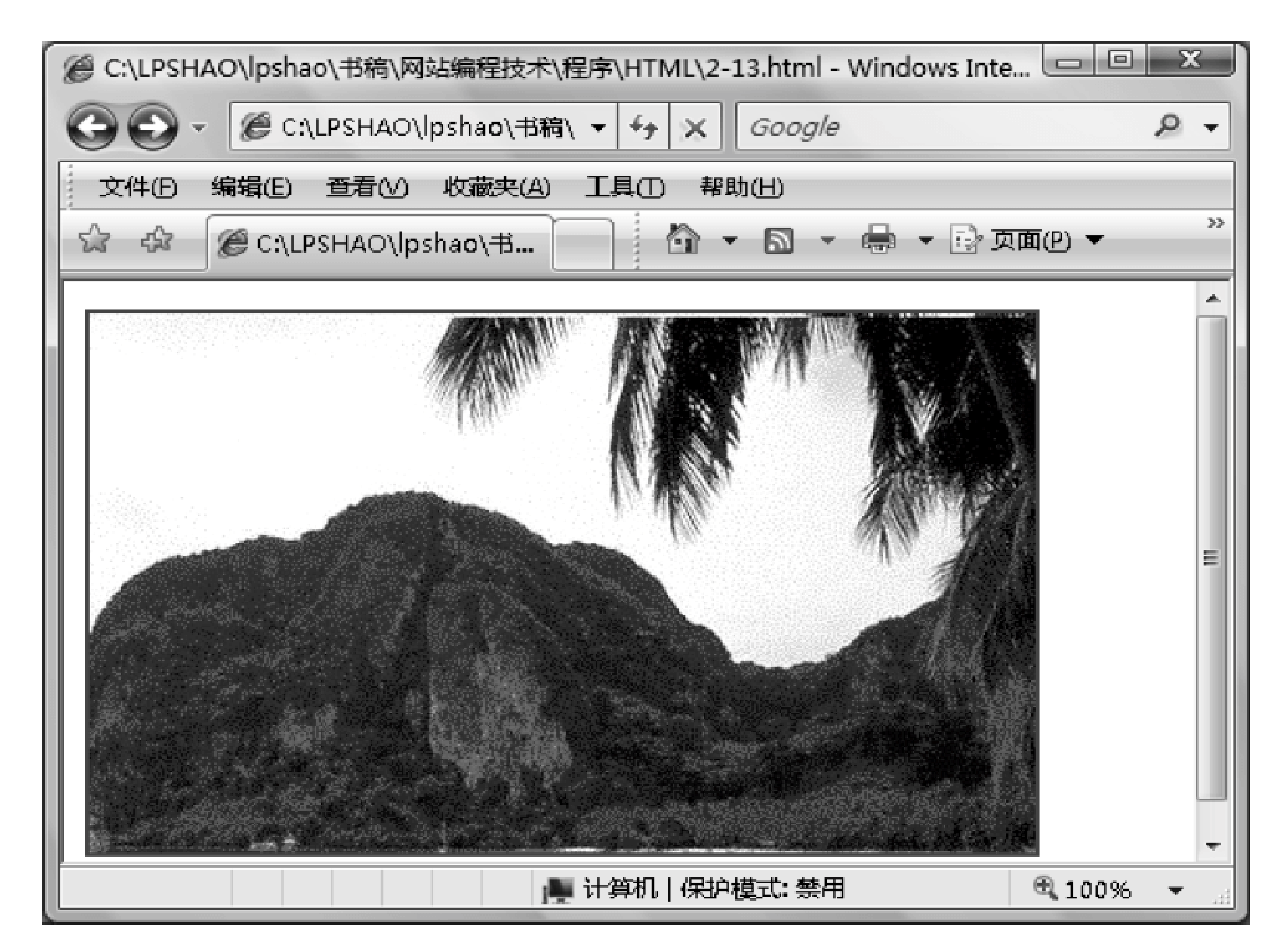


图 2.13　显示地图的页面

2.5　表单及组件标记码

如何通过页面与用户进行交互呢？这需要提供一个让用户输入问题的接口，以及显示服务器对问题给出的答案的接口。form 表单标记码就是解决用户输入问题接口的具体方案。

form 表单标记码可以为用户提供一种交互界面，让用户通过表单组件在页面中输入信息，与其他程序(JSP 或 ASP 程序，JavaScript)一起与用户进行信息交流。form 有两个属性，一是指出其他程序的路径和名称，二是确定提交用户输入的信息给服务器的方式。form 标记码告诉浏览器下面要在页面中显示表单组件。

表单标记码及组件标记码的语法格式：

```
<form method="" action="">
<input type="" name="" value="" size="">
  ⋮
</form>
```

method 属性说明提交给服务器信息时所使用的方式，一般选择 post 或 get。

action 属性说明 form 提交的信息交给服务器中的哪个应用程序进行处理，其路径与

文件名。由于本章不讨论与其他程序的关系，所以下面的 form 没有使用这两个属性。在介绍 JSP 时再使用。

与 form 表单标记码配套使用的是表单组件标记码 input，它用来描述具体的表单组件。

Input 标记码通过 type 属性定义表单组件类型，可以定义 text、password、checkbox、radio、submit、reset 等不同组件。Name 属性定义组件的名称。Value 属性定义组件的默认值。Size 属性定义组件大小。

本节的内容主要解决如何 HTML 文档中添加表单及组件元素，创建与用户进行交互的页面。

2.5.1 文本框组件

如何让用户在页面上输入信息呢？输入简单的信息可以使用文本框组件。

例 2.14 在页面上添加带有初始值的文本框组件的文档(2-14.html 文件)。

```
<html>
<body>
<form>
输入姓名：<input type="text" name=x2 value="lp" size=14><br>
输入口令：<input type="password" name=x3 size=14 maxlengh=18><br>
电话号码：<input type="text" name=x1 value="010 51688529" >
</form>
</body>
</html>
```

2-14.html 显示结果如图 2.14 所示。



图 2.14 添加文本框的页面

可见文本框组件可以让用户输入简短文字。添加文本框组件的方式很简单，只是在<form>标记码对之间 input 标记码中定义 type 属性的值为 text 即可。

通过属性 size 可指定文本框的大小，其值以字符位为单位。如果不指定 size 属性，浏览器以默认值 20 设置 size 的大小。属性 maxlengh 定义最多输入的字符。属性 name 是用来保存通过文本框输入的数据的变量。

属性 value 用来定义文本框的初始值，例如"lp"，需用双引号""。

属性 type 为 password 表示定义的是密码框，用户输入的信息将以字符 • 代替，以免被别人看见。

2.5.2 文本区组件

如果希望用户多输入一些内容，可使用文本区组件。只要在<form>标记码后，插入标记码：<textarea cols= x rows=y>。而后，加上</textarea>和</form>标记码即可，通过 textarea 的 cols 属性可以确定文本框的宽度，定义文本区有多少列。通过 rows 属性可以确定文本区的高度，即有多少行。

例 2.15 在页面上添加一个多行多列文本区的文档(2-15.html 文件)。

```
<html>
<body>
<form>
一个多行多列的文本框<P>
<textarea cols=30 name=x1 rows=8>
</textarea>
</form>
</body>
</html>
```

2-15.html 显示结果如图 2.15 所示。

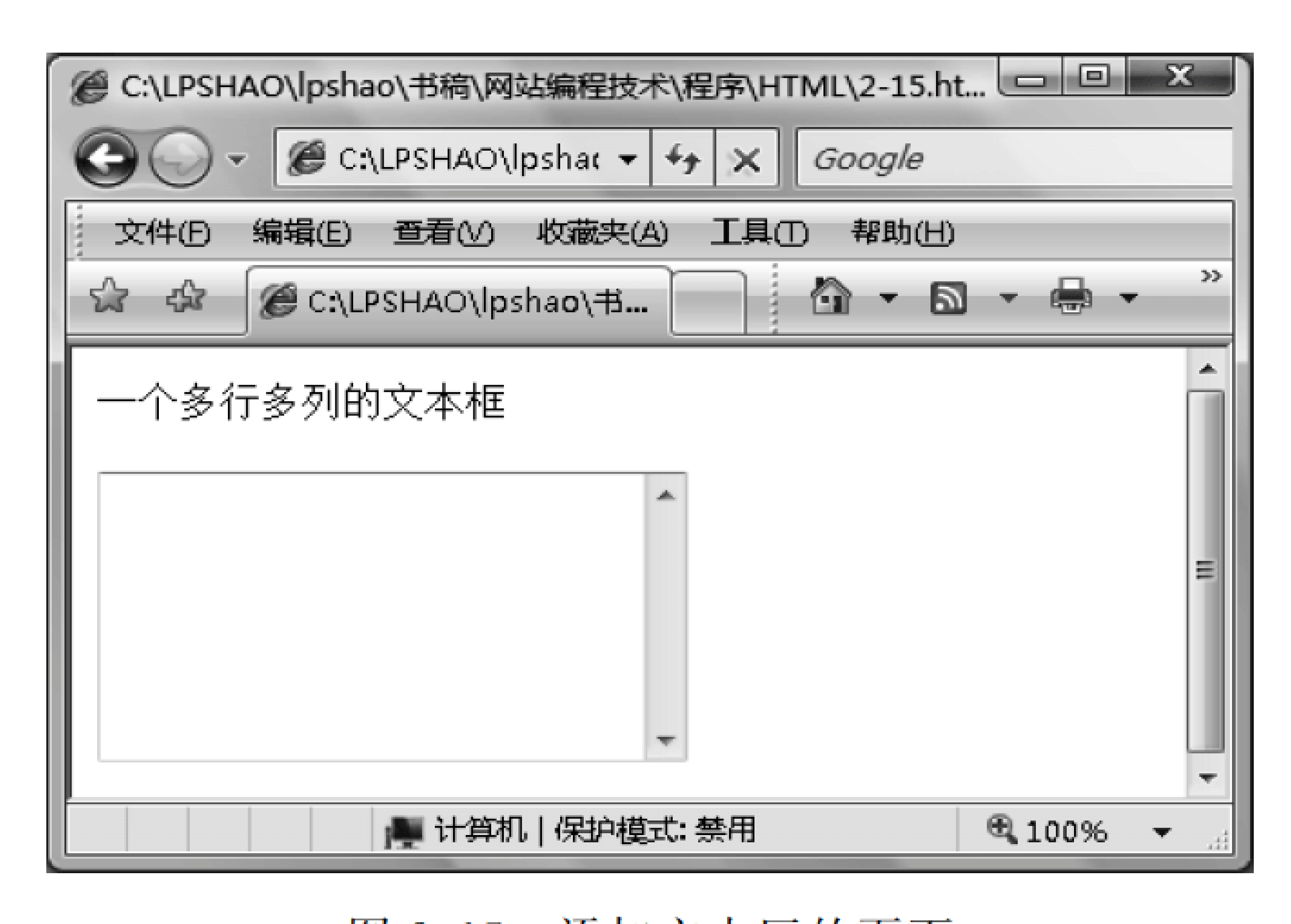


图 2.15 添加文本区的页面

以上的文本框与文本区都不能进行交互工作。如果要使它能进行交互工作，需要使用 JavaScript 文件或 JSP 文件。

2.5.3 提交与重置命令按钮组件

提交(submit)命令按钮的功能是把用户输入的信息提交给相关程序，让服务器或程序进行处理。重置(reset)命令按钮的功能是把用户输入的内容从文本框或文本区清除

掉,以便重新输入。

在页面上添加这两个命令按钮与添加文本框组件类似,在表单组件标记码下面再加入<input type="submit">和<input type="reset">即可。通过 input 标记码的 value 属性可以定义命令按钮上显示的文本,默认文本为 submit 和 reset。一般情况下,form 标记码对都要具有文本框、文本区等输入信息的组件与提交及重置命令按钮组件,这样表单才能具备完整的处理功能。

例 2.16 在页面添加提交及重置命令按钮表单组件的文档(2-16.html 文件)。

```
<html>
<body>
<form>
一个多行多列的文本框<P>
<textarea cols=30 Name=x1 rows=8>
</textarea>
<input type="submit" value="提交">
<input type="reset" value=" 重置">
</form>
</body>
</html>
```



图 2.16 添加命令按钮的页面

2-16.html 显示结果如图 2.16 所示。

2.5.4 单选框与复选框组件

单选框(radio)让用户通过选择输入信息,更方便用户输入。通过单选框一次只能从多个选项中选择一项。在页面上添加单选框,只需在<form>标记码后加上<input type="radio">,并定义其 name、value 属性,再以</form>结束即可。

创建复选框(checkbox)是为了方便用户选择多个选项内容。添加一个多选框只要在<form>标记码后加上<input type="checkbox">,并定义其 name、value 属性,再以</form>标记码结尾即可。

例 2.17 在页面添加单选框和多选框组件的文档(2-17.html 文件)。

```
<html>
<body>
<form>
<h3>单选框
<input type="radio" name="x1" value="音乐" checked>音乐
<input type="radio" name="x1" value="电影">电影
<input type="radio" name="x1" value="小说">小说
<input type="radio" name="x1" value="电视">电视</h3>
<input type="submit" value="提交">
<input type="reset" value="重置">
<h3>多选框</h3><input type="checkbox" name="x1">北京
```

```
<input type="checkbox" name="x2">上海
<input type="checkbox" name="x3">天津
<input type="checkbox" name="x4">重庆
<input type="checkbox" name="x5" checked>武汉(已被选择)<P>
<input type="submit" value="提交">
<input type="reset" value="重置">
</form>
</body>
</html>
```

2-17.html 结果显示如图 2.17 所示。

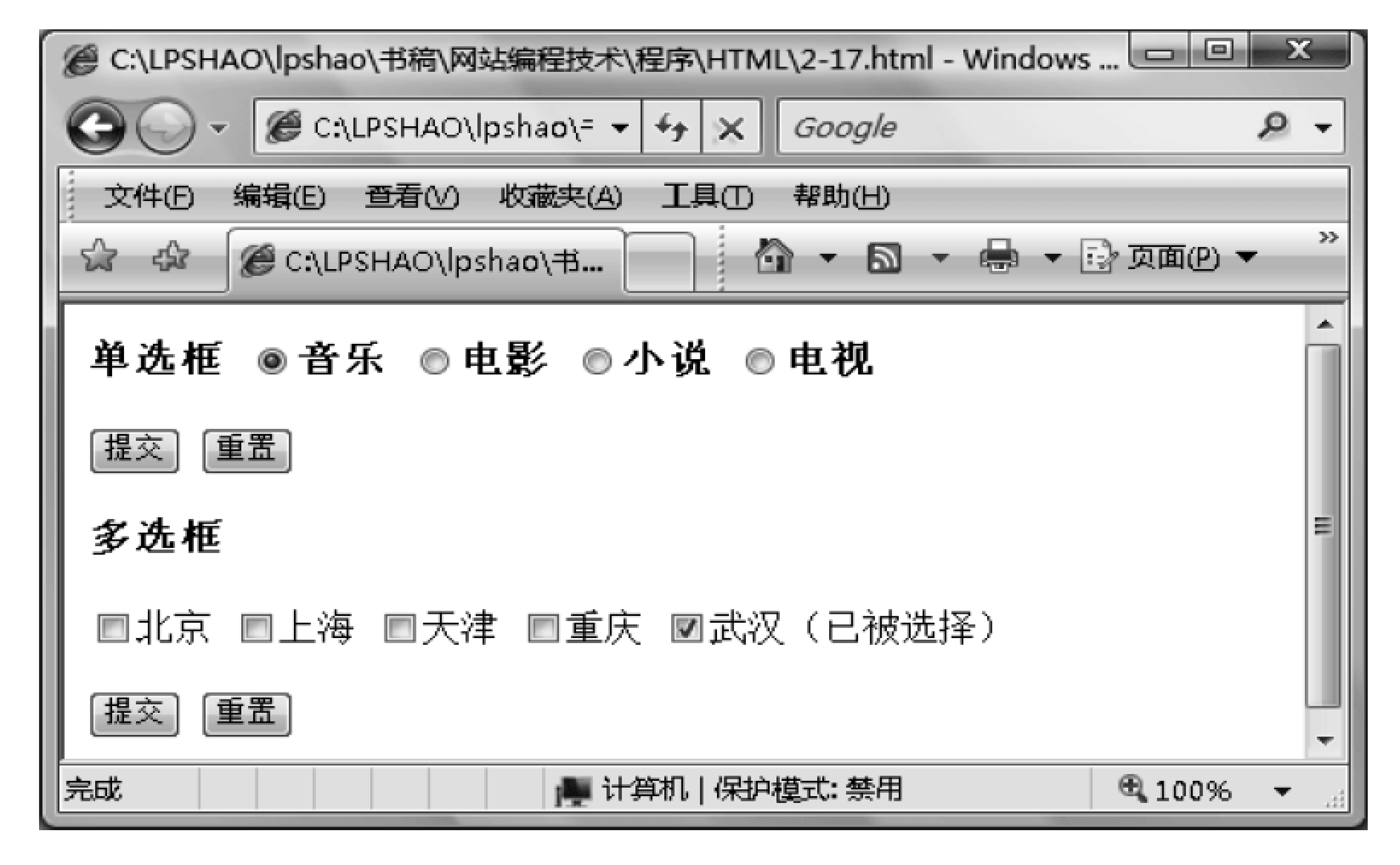


图 2.17 添加单选框与复选框的页面

注意:单选框 name 只能选取一个名称,所以属性值都是 x1,只能保存一个选项结果。如果想让其中的某个按钮为默认值,只要在 input 标记码中加入 checked(已选择)属性即可。复选框中 name 可以选取不同的名称 x1,x2,x3,…,保存不同选项结果。

2.5.5 下拉菜单与滚动菜单组件

下拉菜单与滚动菜单的功能与单选框类似,也是方便用户通过选择输入信息。

下拉菜单以<form>标记码开始,使用<select>标记码告诉浏览器显示一个给定菜单选项的下拉菜单组件。通过<option>标记码与菜单选项文本给出下拉菜单的菜单内容。下拉菜单组件的宽度默认情况下由最长的选择项确定。

滚动菜单只是在下拉菜单<select>标记码中加入 multiple 属性,允许用户一次选择多个选项,类似复选框如<select multiple size=x>。size 属性定义显示在菜单里的选择项(或选择行)数。滚动菜单的宽度同样由最长的选择项的宽度决定。

例 2.18 包含下拉菜单与滚动菜单组件的文档(2-18.html 文件)。

```
<html>
<body>
<form>
```

```
下拉菜单中任选一个
<select name=x1>
<option>北京汽车站
<option>上海汽车站
<option>武汉汽车站
<option>重庆汽车站
</select>
<P><input type="submit" value="提交">
<input type="reset" value="重置">
<P>滚动菜单可多选<P>
<select name=x1 multiple size=5>
<option>苹果
<option>香蕉
<option>葡萄
<option>樱桃
<option>草莓
<option>梨
</select>
<p><input type="submit" value="提交">
<input type="reset" value="重置">
</form>
</body>
</html>
```

2-18.html 显示结果如图 2.18 所示。

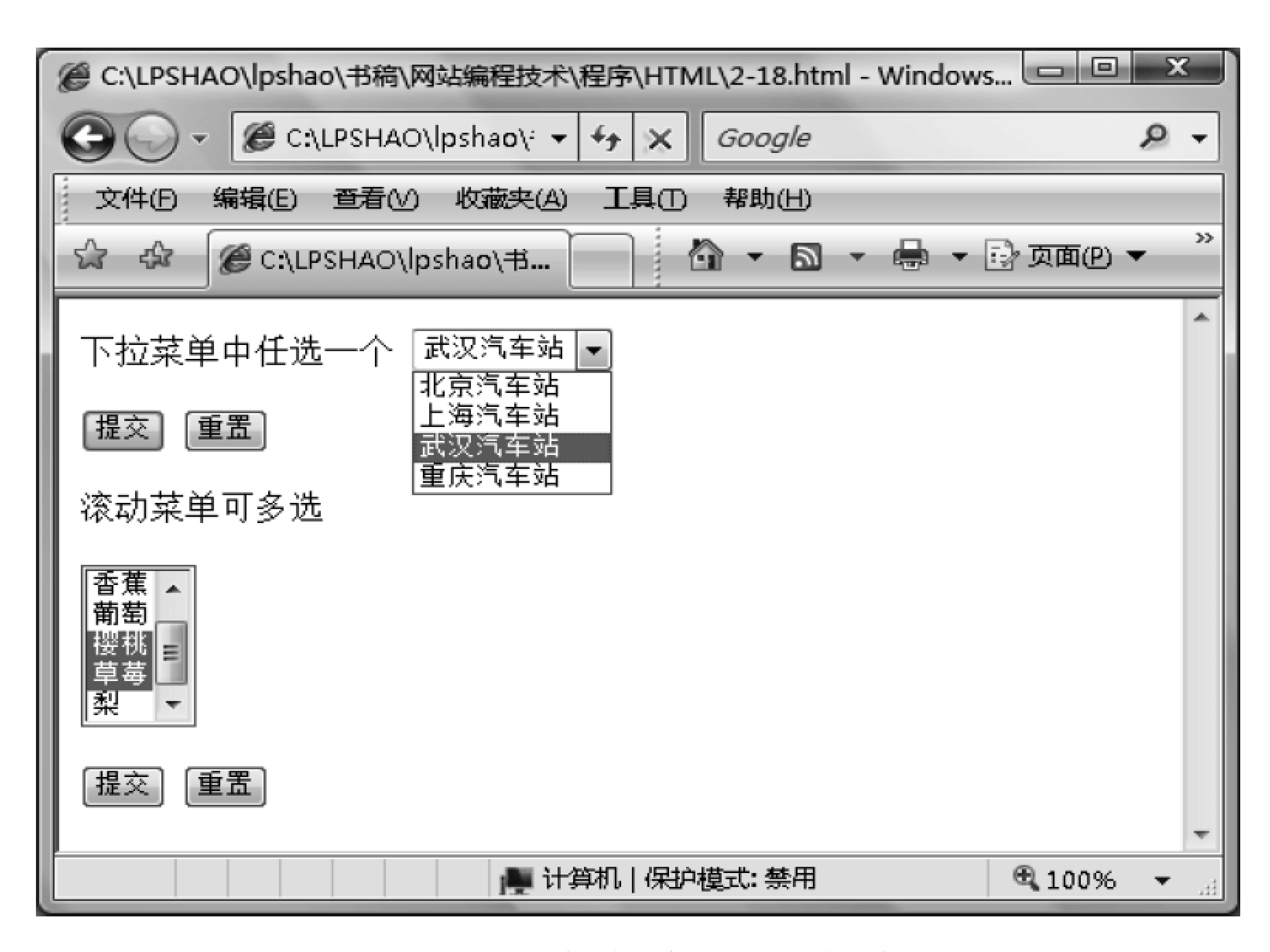


图 2.18 添加下拉菜单与滚动菜单的页面

注意：要添加输入信息的表单组件必须以<form>开始，以</form>结束。一个页面可以添加多个表单标记码，每当新加入一个<form>标记码，浏览器就产生一个新的段落。

可以将多个 form 组件放在同一对<form></form>标记码中，而无须添加一个组件元素就放上一对<form></form>标记码。

HTML 语言的标记码就介绍到这里，虽然只讲了一部分标记码，但编写 html 文档已经够用了，其他内容可以在实践中继续学习。现在可以欣赏别人的 HTML 网页，也可以自己编写简单的 HTML 网页了。

学到这里是不是觉得 HTML 语言编程的技术很简单呢？实际上，只要记住两个原则：一个是以<HTML>开始和以</HTML>结束，再一个就是在<标记码>(前面的开始标记码)中添加属性参数，属性参数虽然看起来挺多，但许多是通用的。如果有兴趣想了解更多的 HTML 标记码，可继续自学下去，因为还不断会有新的标记码出现。

2.6 思考与练习

2.6.1 思考题

2-1 什么是 HTML 语言？它有什么特点？

2-2 HTML 文档的主要作用是什么？

2-3 HTML 语言有哪些类型的基本标记码？

2-4 地图标记、表单标记、框架标记有什么相同处？它们与其他标记有什么不同处？

2.6.2 上机练习

2-1 使用 Word 编写一个 HTML 文件，存为 2-1.html，并在浏览器中运行一遍。

2-2 使用“附件”的“记事本”编写一个 HTML 文件，存为 2-2.html，并在浏览器中运行一遍。

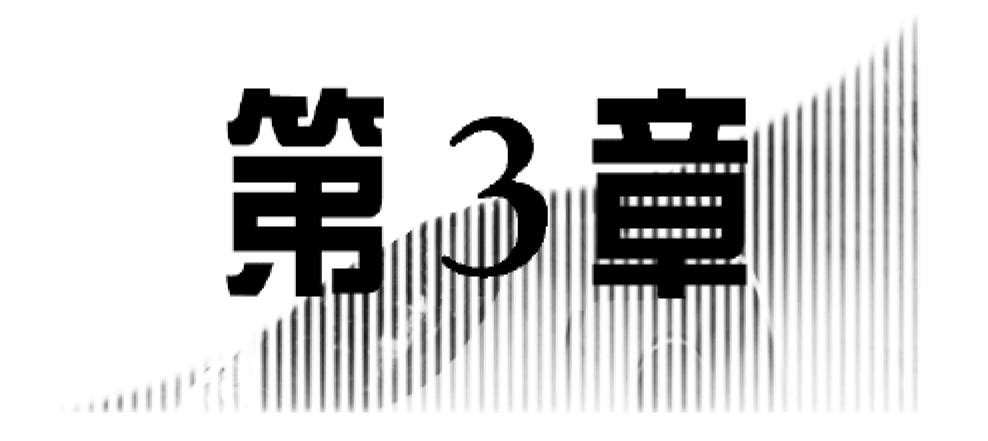
2-3 使用 FrontPage 工具编写或编辑一个 HTML 文件，存为 2-3.html，并在浏览器中运行一遍。

2-4 编写一个带有表格、图片和文字的页面，并添加背景颜色和背景音乐。

2-5 编写带有不同超链接的 HTML 文件。

2-6 编写一个带有 map 标记码的页面。

2-7 编写一个带有表单的页面，包括文本框、口令框、单选框、下拉菜单组件。



第3章 脚本语言——JavaScript

使用 JavaScript 脚本语言，可使 HTML 网页具有简单的动态性与计算功能。JavaScript 可以嵌入到 HTML 文档与超文本标识语言一起显示页面信息，实现在页面中与用户的交互功能，开发能在客户端完成的 Web 应用程序。

本章主要解决以下问题：

- 什么是 JavaScript；
- 如何将 JavaScript 脚本代码嵌入到 HTML 文档；
- 什么是 JavaScript 对象；
- 如何使用 JavaScript 对象的属性与方法；
- 如何定义 JavaScript 变量；
- 如何创建 JavaScript 函数；
- 如何控制 JavaScript 语句的流程；
- 什么是 JavaScript 事件与事件处理方法；
- 如何编写与使用单独保存的 JS 文件。

3.1 在 HTML 中插入 JavaScript

本节主要解决什么是 JavaScript、如何将 JavaScript 脚本代码嵌入到 HTML 文档、什么是 JavaScript 对象，如何使用 JavaScript 对象的属性与方法等问题。

3.1.1 什么是 JavaScript

JavaScript 是 Java 的子集，具有 Java 语言的基本语句，具有 Java 语言的基本特性，是由对象、对象属性与对象方法(函数)构成的，是一种基于对象和事件驱动、具有安全性的脚本语言，其主要特点是比 Java 语言容易理解，能与 HTML 语言一起完成显示页面信息的任务。

可能有人会问，运行 JavaScript 需要新的软件环境支持吗？回答是不用，与 HTML 语言相同，还是使用 Web 浏览器。目前大部分的 Web 浏览器都支持 JavaScript 的功能。

3.1.2 在 HTML 文档中插入 JavaScript 语言的方式

1. 将 JavaScript 代码插入到 HTML 文档

例 3.1 一个包含有 JavaScript 语言的 HTML 文档(3-1.htm)。

```
<html>
<body>
<script language="JavaScript">
/* 隐藏下面 JavaScript 语句
多行注释
多行不显示
多行不显示的情况 */
document.write("<h4>这是我们第一次使用 JavaScript 语言</h4><hr>")
//隐藏一行
</script>
</body>
</html>
```

保存文件为 3-1.htm,在浏览器打开 3-1.htm 网页,可以看到如图 3.1 所示页面。



图 3.1 网页 3-1.htm 的运行结果

2. 知识点归纳

1) 插入 JavaScript 语言的<script></script>标记码

从 3-1.htm 中可以看出 JavaScript 语言是通过<script></script>标记码对插入到 HTML 文档中的,而且可以放在 HTML 文档中的任何地方。它以<script language="JavaScript">开始,以</script>结束。

插入 JavaScript 语言的语法格式:

```
<script language="JavaScript">
  JavaScript 语句
</script>
```

2）JavaScript 命令的大小写之别

JavaScript 与 Java 命令一样区别大小写。因此，在 3-1.htm 中的 JavaScript 命令代码都是小写字母，如例子中的 document.write( )语句。

3）JavaScript 的注释标记

JavaScript 有自己的注释标记。用来告诉浏览器将 JavaScript 语句隐藏起来不显示。

JavaScript 注释标记的语法格式：

```
//一行注释
/*多行注释*/
```

说明：// 标记注释一行，即一行不显示，而/*…*/标记可以将多行隐藏起来不显示。

4）document 对象及 write 方法

document 对象为 JavaScript 内置的文档对象，它具有 write()方法，可以在页面中显示指定的文字。

3.1.3 使用 JavaScript 对象属性与方法

1. 创建一个包含多个 JavaScript 对象的文档

例 3.2 一个使用 JavaScript 的 document 对象的 lastModified 属性显示最后修改日期、使用 window 对象的 alert 方法显示一个提示对话框、使用 window 对象的 prompt()方法显示一个输入信息对话框的文档(3-2.htm)。

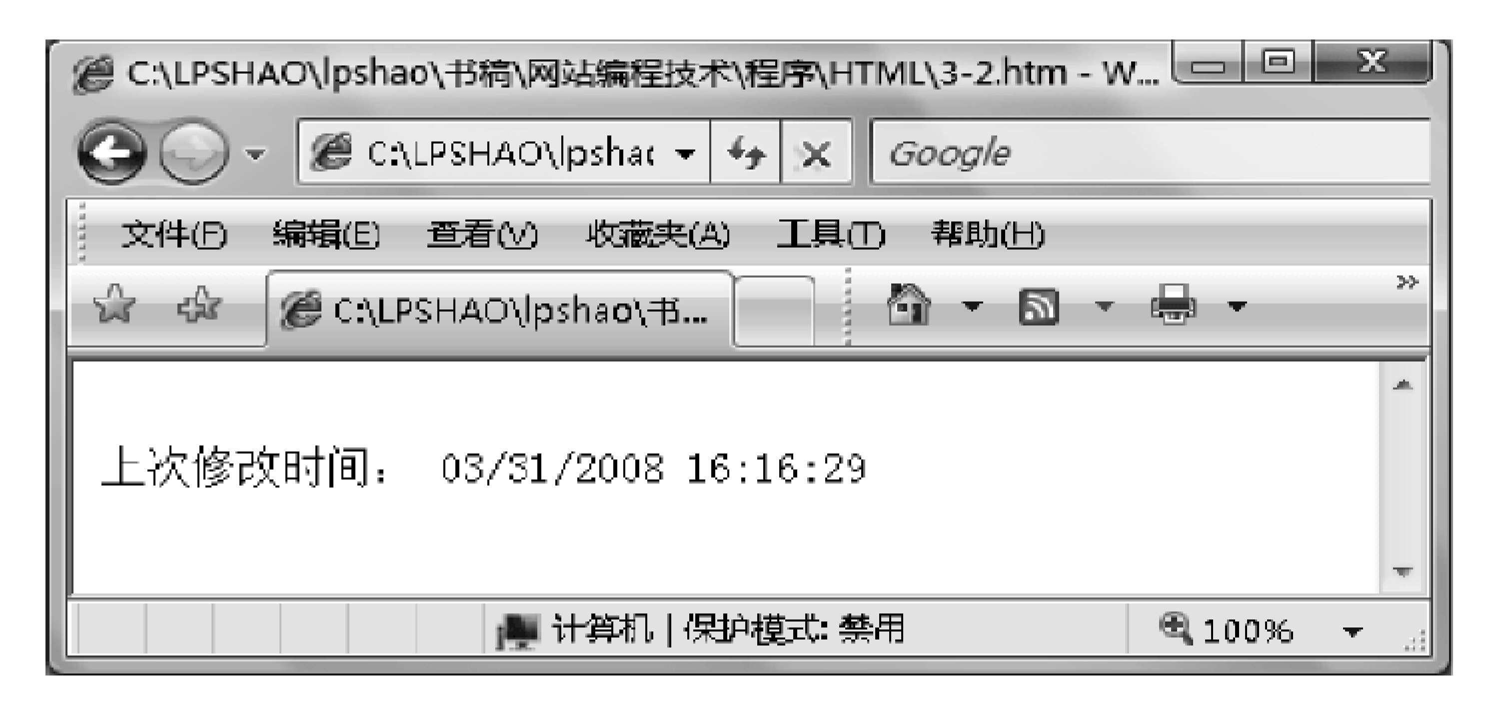
```
<html>
<body>
<br>上次修改时间：
<script language="JavaScript">
document.write(document.lastModified)
alert("欢迎你第二次使用 JavaScript")
prompt("请输入你的姓名:","张驰")
</script>
</body>
</html>
```

在浏览器打开 3-2.htm 网页，可以看到如图 3.2 所示页面。

2. 知识点归纳

1）调用对象方法和属性的方式

在 3-2.htm 中使用了 document 对象的 write()方法和 lastModified 属性。在 document.write()语句点(.)左边是“document”对象名，点(.)右边 write 为 document 对象的方法名，write()方法的功能是输出圆括号内的参数值到窗口。JavaScript 语言通过对象定义其方法，通过对象名.方法名来调用对象的方法完成指定的功能。例如通过



(a) 网页 3-2.htm 的运行结果



(b) 提示对话框



(c) 输入信息对话框

图 3.2　例 3.2 程序运行结果

document.write()调用了 document 对象的 write()方法。write()方法完成显示信息的任务，显示的信息由圆括号对内的参数提供。

调用对象方法和使用对象属性有如下语法格式：

```
对象名.方法名 ([参数])
对象名.属性名
```

2）alert 方法的功能

在 3-2.htm 中还使用了 window 对象的 alert 方法，它的功能是显示一个带有提示信息的对话框，如图 3.2 所示。alert()方法一般用于提示或提醒用户某一事情。

调用 alert 方法的语法格式：

```
alert("提示信息")
```

提示对话框中的提示信息由 alert()括号内的“提示信息”定义。单击对话框“确定”按钮，对话框会消失。

alert()方法前面没有对象名，这是因为 alert()方法为 window 对象的方法，没有指明对象名，JavaScript 默认为 window 对象。

3）prompt()方法的功能

在 3-2.htm 中还使用了 window 对象的 prompt()方法，它的功能是显示一个可以让用户输入信息的对话框。输入信息对话框由“提示信息”，“初值”、一个“确定”按钮和一个“取消”按钮组成，单击“确定”按钮可以提交用户输入的信息，同时关闭对话框。

调用 prompt()方法的语法格式：

```
prompt("提示信息", "初值")
```

prompt()方法的括号中包括了两个字符串，一个是要显示的信息，一个是输入文本框内显示的初始信息，本例中为“张驰”，也可以什么都不写或写成任意字符。所用的引号一定要成对使用，不能是一个双引号一个单引号。引号一定为西文的。

3.2 JavaScript 变量、数据类型与表达式

本节主要解决如何声明保存数据的 JavaScript 变量，有哪些类型的 JavaScript 数据，有哪些类型的数据运算符，有哪些类型的表达式等问题。

3.2.1 声明 JavaScript 变量的方式

1. 变量与常量

在例 3.2 中使用对话框输入的消息都丢失了，怎样才能保存用户在页面上输入的信息呢？可以使用变量。

变量是计算机提供的存放数据的内存空间的代号。变量的功能就是保存数据，存放在变量中的数据可以改变，也就是说变量可以保存不同的数据。

常量是计算机中存放固定不变数据的内存空间的代号。

在 JavaScript 中使用变量和常量前要先声明其代号，即变量名，如图 3.3(a)所示，在应用程序中通常用拼音字母或英文单词作为变量名(变量的代号)。

常量在使用过程中存放的是固定不变的数据，如图 3.3(b)所示。

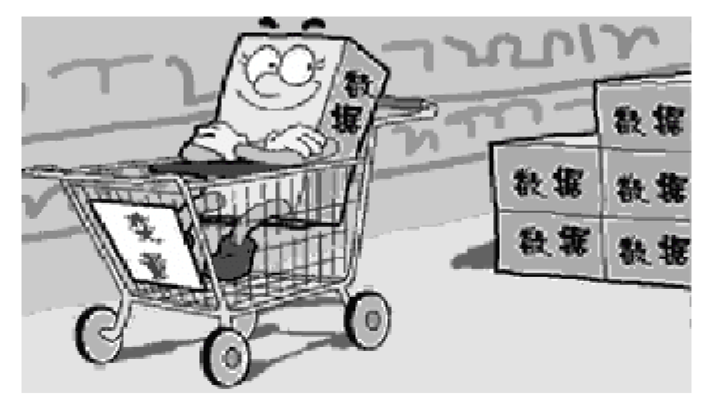
变量在使用过程中可以不存放数据(为空)，也可以存放不同数据，如图 3.3(b)与(c)所示。



(a) 数据常量



(b) 空变量



(c) 有数据的变量

图 3.3 变量与常量

2. 在程序中声明变量

例 3.3 带有 JavaScript 变量，并给变量赋值的文档(3-3.htm)。

```
<html>
<body>
<script language="JavaScript">
var name=prompt("请输入你的姓名:","张驰")
var x=7
```

```
var y=8
document.write("欢迎你,"+name)
document.write("<p>x+y="+(x+y))
</script>
</body>
</html>
```

在浏览器打开 3-3.htm 网页,可以看到如图 3.4 所示页面。



图 3.4 网页 3-3.htm 的运行结果

在图 3.4 所示的输入对话框输入 slp,单击"确定"按钮,在浏览器中可以看到如图 3.4 所示的结果。可以输入任意一个名字,会看到输入的名字显示在页面上。

3. 知识点归纳

1) 声明变量的方法

JavaScript 声明变量的方法和变量赋值的语法格式:

var 变量名=数据值

JavaScript 的 var 关键字用来声明一个 JavaScript 变量,并将"数据值"赋给变量。例如,在 3-3.htm 中用 var 定义了 3 个变量 x、y 和 name,并将 7,8 数据值赋给了变量 x、y。另外将 prompt()方法得到的数据"slp"赋给了变量 name。给变量赋值所用的等号(=)称为赋值运算符。

2) 选取变量名的规则

变量名对大小写非常敏感,大小写字母代表不同的变量。另外,变量名称的长度是任意的,但必须遵循以下规则:

- 第一个字符必须是一个字母(大小写均可)、或一个下划线(_)或一个美元符($)。
- 后续的字符可以是字母、数字、下划线或美元符。
- 变量名称不能是 JavaScript 的关键字。

变量可以用来保存数据,JavaScript 中的数据指的是什么呢? 数据有什么不同吗?

3.2.2 JavaScript 的数据类型

JavaScript 有 6 种数据类型。不同类型的数据可以表示不同的信息,进行不同的数据计算。

1. 数值数据类型

数值数据类型包括整型 int,例如 12、22。整型数可以为正数、0 或者负数;浮点型 float,例如 7.254、－3.4、2E5,浮点数可以包含小数点、也可以包含一个 e(大小写均可,在科学记数法中表示“10 的幂”)、或者同时包含这两项。

2. 字符串类型 string

例如“你好!”、“Hello!”、“254”都是 string 类型的数据。字符串数据必须使用单引号或双引号括起来。

3. 逻辑类型 boolean

boolean 数据类型的值只有两个:true 与 false。这是两个特殊值,不能使用 1 和 0 代替。

4. 空数据类型 null

null 数据类型只有一个值 null,表示没有任何值,它表示什么也没有。空 null 与 0 值或空字符串""是不一样的,0 值或空字符串""是一个实际值。

5. undefined 未定义数据类型

undefined 数据类型只有一个值 undefined,表示变量被创建后未给该变量赋值。如果对象属性不存在,声明了变量但从未赋值,将返回 undefined 值。例如可使用语句 if(typeof(x) == "undefined")判别变量 x 是否赋值。

6. Object 对象类型

除了上面提到的各种常用类型外,JavaScript 中包含对象类型。表示数据为某种对象(类)。

变量和常量也具有数据类型。变量的数据类型根据其赋值的数值的数据类型来确定。

3.2.3 JavaScript 的运算符

运算符即对数据进行运算的符号,JavaScript 按照运算元素的个数可分为一元、二元和三元 3 种运算符,一元运算符只有一个运算元素,如 x++ 或 ++x;二元运算符有两个运算元素,如 5 * 4;三元运算符具有三个运算元素。

按照运算符的运算规律来划分有如下 6 种运算符:

1. 赋值运算符

赋值运算符有如下 6 种类型：

(1)“＝”表示将右边数赋值给左边变量。例如 X＝5，5 被赋给了 X。

(2)“＋＝”表示左、右两边数相加，结果赋给左边变量。例如 X＝5，Y＝5，X＋＝Y 将 10 赋给了 X。

(3)“－＝”表示左、右两边数相减，结果赋给左边变量。例如 X＝5，Y＝5，X－＝Y 将 0 赋给了 X。

(4)“＊＝”表示左、右两边数相乘，结果赋给左边变量。例如 X＝5，Y＝5，X＊＝Y 将 25 赋给了 X。

(5)“/＝”表示右边数除左边数，结果赋给左边变量。例如 X＝5，Y＝5，X/＝Y 将 1 赋给了 X。

(6)“％＝”表示右边数除左边数，余数赋给左边变量。例如 X＝25，Y＝7，X％＝Y 将 4 赋给了 X。

2. 算术运算符

算术运算符有加“＋”、减“－”、乘“＊”、除“/”、取余“％”和取负 X＝－X、加 1“＋＋”，减 1“－－”等。

例如 Y＝＋＋5，Y 的值为 6；例如 Y＝－－6，Y 的值为 5。其他运算符的含义与平时的数值运算一样，例如 X＝54＋5＊2，X 的值为 44。

3. 字符串运算符

字符串运算符只有一个合并运算符“＋”，表示将两个字符串合并在一起，例如：'abc'＋"你好！"，合并后的值为“abc 你好！”。

4. 比较运算符

比较运算符有如下 6 种类型：

(1)“＝＝”：如果两边数据相等返回 true。例如：2＝＝2 返回 true。

(2)“！＝”：如果两边数据不相等返回 true；例如：“THE”！＝“TH”返回 true。

(3)“>”：如果左边数大于右边数返回 true；例如：2>1 返回 true。

(4)“<”：如果左边数小于右边数返回 true。

(5)“>＝”：如果左边数大于或等于右边数返回 true。

(6)“<＝”：如果左边数小于或等于右边数返回 true。

5. 逻辑运算符

逻辑运算符有如下 3 种类型：

(1)“&&”：称为逻辑与，当表达式两边的逻辑值都是 true 时结果为 true，否则结果为 false。例如：2>1 && 5<4，其结果为 true；2>1 && 5>4，其结果为 false。

(2) “||”：称为逻辑或，当表达式两边的逻辑值中任意一个是 true 时结果为 true，当表达式两边的逻辑值都为 false 时结果为 false。例如，2>1 || 5<4，其结果为 true。2>1 ||5>4，其结果也为 true。

(3) “!”：称为逻辑非，当表达式的逻辑值为 true 时返回结果 false，当表达式的逻辑值为 false 时返回结果 true。例如：!2>1 结果为 false。

6. 条件运算符

条件运算符只有一个“?”，条件运算符与上面的运算符略有不同。条件表达式的结构如下。

```
(条件)? 结果 1:结果 2
```

首先计算条件（逻辑表达式或比较表达式）的值，值为 true 表达式的值为结果 1，值为 false 表达式的值为结果 2。例如，条件表达式如下：

```
(day= = "星期六")?"今天是周末":"今天不是周末"
```

当变量 day 为“星期六”时表达式的结果为“今天是周末”，否则结果为“今天不是周末”。

条件运算符为三元运算符。

7. 运算符优先级

当表达式里有多种运算符号的时候，运算的次序是什么呢？其优先级从高到低顺序为：

(1) 乘、除、模(* 、/、%)；

(2) 加、减(+、-)；

(3) 比较(<、>、<=、>=、==、!=)；

(4) 逻辑与(&&)；

(5) 逻辑或(||)；

(6) 条件(?)。

优先级指的是复合运算进行计算时的先后顺序，对于所有的二元运算都是从左到右进行计算，可以用圆括号()来忽略优先级，即()小括号优先级最高。例如：

- 算术表达式(5+4) * 5 的值为 45，而 5+4 * 5 的值为 25。
- 逻辑表达式 false||true&&false 的值为 false；
- 条件表达式'4 * 5+18>22'的值为“对”。

3.2.4 JavaScript 的表达式

1. JavaScript 表达式

JavaScript 表达式是 JavaScript 变量、常量运算符以及表达式的任意组合。JavaScript 表达式的计算结果为一个数据值，表达式的类型确定其值的数据类型。

2. JavaScript 表达式的类型

根据组合表达式运算符的类型，JavaScript 的表达式可分为 6 种类型：赋值表达式、算术表达式、字符串表达式、比较表达式、逻辑表达式和条件表达式。

例 3.4 使用 JavaScript 表达式，给出表达式结果的文档(3-4.htm)。

```
<html>
<head>
<script language="JavaScript">
var question="10+20是多少？";
var answer=30;
var x1="计算正确";
var x2="计算错了";
var x5=prompt(question,"0");
var output=(x5==answer)?x1:x2;
</script>
</head>
<body>
<script language="JavaScript">
document.write(output);
</script>
</body>
</html>
```

在浏览器打开 3-4.htm 网页，可以看到如图 3.5 所示对话框，输入 30 并单击“确定”按钮后，会出现图 3.5 所示页面。



(a) 输入对话框



(b) 网页 3-4.htm 的运行结果

图 3.5 例 3.4 程序运行结果

3. 知识点归纳

1）条件表达式的使用方式

在 3-4.htm 里使用了条件表达式((x5==answer)? x1:x2)，并将结果值赋给了变量 output，其中变量 x5 的值是由用户输入的。

2）定义变量在 HTML 文档的位置

3-4.htm 中把变量的定义都放在了 HTML 文档的头部，把输出的结果放在 HTML 文件的主体中，这样写代码可以使程序清晰，容易看懂。这个习惯可看成是一个书写 JavaScript 代码的规则。

3）JavaScript 命令行的结束标志(;)

在 3-4.htm 中每个 JavaScript 命令行都添了加分号(;)，一行可以写多个 JavaScript 命令，只要用分号(;)分开即可。为了统一起见，本书在 JavaScript 的每个命令行都使用了分号(如果一行只有一个 JavaScript 命令语句，不加分号(;)也是可以的)，但要注意必须是西文方式的分号(;)，代码中其他的符号也要用西文符号，否则会出错。

4）在 HTML 文档插入 JavaScript 语句的次数与位置

在 3-4.htm 中可以看到插入了两次 JavaScript 语句。根据需要在 HTML 文件中可以多次插入 JavaScript 语句，而且可以在任何地方插入。

3.3 JavaScript 对象实例与函数

本节主要解决如何创建一个对象实例，如何通过对象实例调用对象的方法与属性，如何创建 JavaScript 的自定义函数等问题。

3.3.1 创建 JavaScript 的对象实例

1. 什么是对象实例

在 JavaScript 中有许多已定义好的内置对象，如窗口对象 window、文档对象 document、日期对象 Date 等。

什么是对象实例呢？为什么要创建对象实例呢？

对象实例即对象的一个具体例子，可以用任意名字来定义对象实例。对象实例是一个具体的对象，它具有原对象的一切属性和方法，通过对象实例还可以定义新的属性和方法。有的对象的属性与方法必须通过对象实例才能调用。

2. 在程序中创建对象实例

例 3.5 包含 Date 对象实例，显示计算机的当前日期和时间的文档(3-5.htm)。

```
<html>
<body>
<script language="JavaScript">
today=new Date()
```

```
document.write("现在的时间是:",today.getHours(),":",today.getMinutes())
document. write ( " < br > 今 天 的 日 期 是:", today. getMonth ( ) + 1,"/", today.
getDate(),"/",today.getYear())
</script>
</body>
</html>
```

在浏览器打开 3-5.htm 网页,可以看到如图 3.6 所示页面。

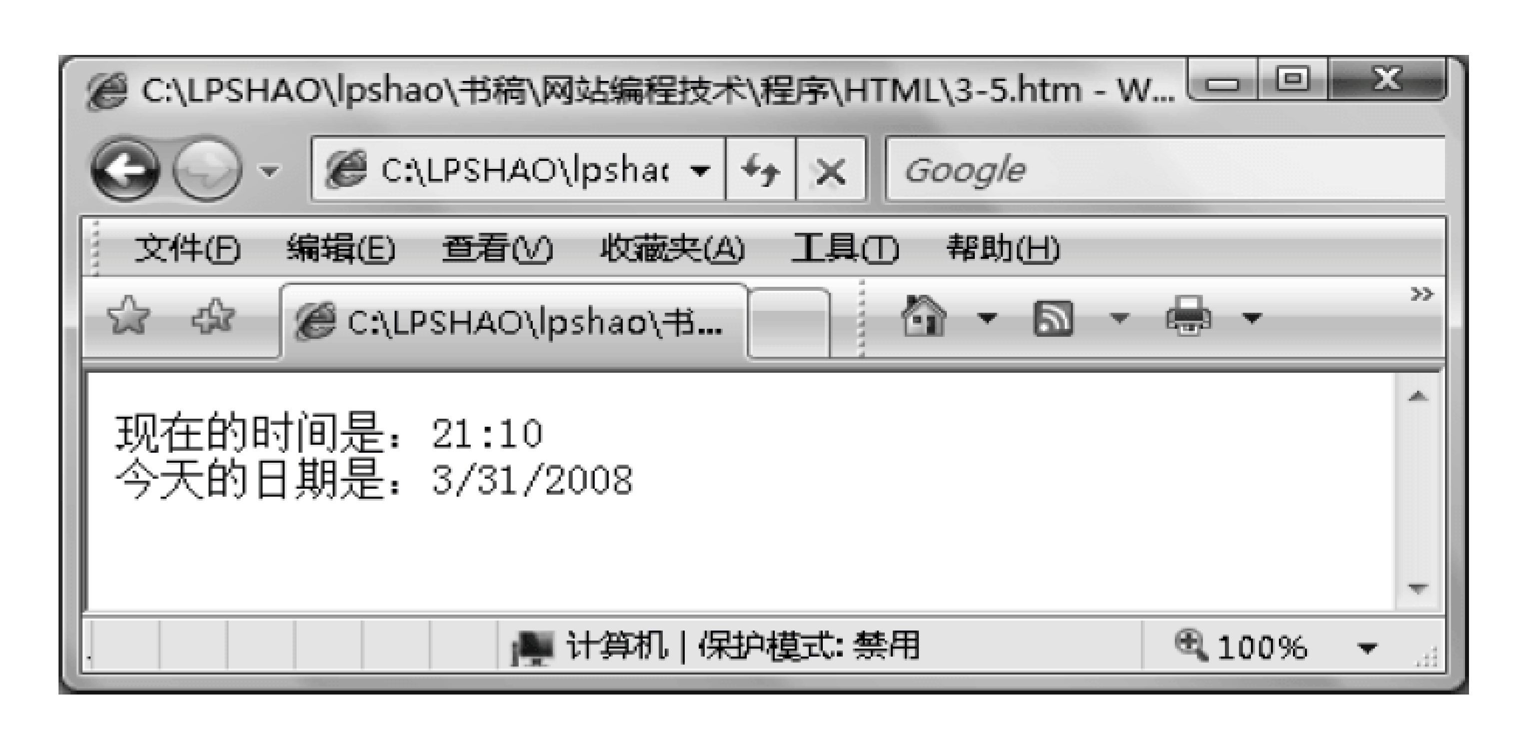


图 3.6　网页 3-5.htm 的运行结果

3. 知识点归纳

1) 创建对象实例的方式

在 3-5.htm 中,Today 为日期对象 Date 的对象实例,它继承了 Date 对象的所有属性和方法。使用 new 语句可以建立一个对象实例。

创建对象实例的语法格式:

```
实例名=new 对象名 ( )
```

2) 通过对象实例调用原对象的属性与方法

在 3-5.htm 中 Date()的对象实例 today 能调用 Date()的属性与方法,在创建对象实例 today 时如果未在 Date()中指定具体日期与时间,将使用本地计算机的当前时间初始化实例对象,因此,通过 today 对象的 get…方法可以得到当前时间与日期。

注意 getMonth 属性所取得的月份范围是从 0 到 11,必须加 1 以才能代表现实中的 1 月至 12 月。

在创建对象实例 today 时,如果 Date()指定一个具体日期,如:date1 = new Date(2008,08,15),使用 today 的 get…方法将显示 2008 年、8 月、15 日。同样,初始化时可加上时间的定义,如:date2 = new Date(2008,08,15,10,50,50)前三个是日期的年、月、日,后三个是时、分、秒。

3.3.2　创建 JavaScript 的自定义函数

1. 什么是 JavaScript 函数

什么是 JavaScript 函数?为什么要使用函数呢?

JavaScript 的函数是执行某个指定任务或功能的一系列 JavaScript 命令的集合。

通过调用一个函数名能方便地实现函数中定义的功能。通过调用函数名可以重复执行某种任务和活动。JavaScript 已经定义好一些内置函数,在编写 JavaScript 脚本时可以直接使用。

这里要介绍的是根据任务和活动的具体内容自定义函数的方式。

JavaScript 的函数与 JavaScript 的方法相似,不同之处是方法与某个对象有关系,通过对象才能使用其方法。而函数与任何对象都没有关系,可直接调用。

2. 在程序中创建自定义函数

例 3.6 一个包含产生随机数的自定义函数的文档(3-6.htm)。

```
<html>
<head>
<script language="JavaScript">
function RandomNumber() {
  today=new Date();
  num=Math.abs(Math.sin(today.getTime()));
  return num;
}
</script>
</head>
<body>
<script language="JavaScript">
  document.write("这是自定义函数 RandomNumber()产生的随机数:",RandomNumber());
</script>
</body>
</html>
```

在浏览器打开 3-6.htm 网页,可以看到如图 3.7 所示页面。

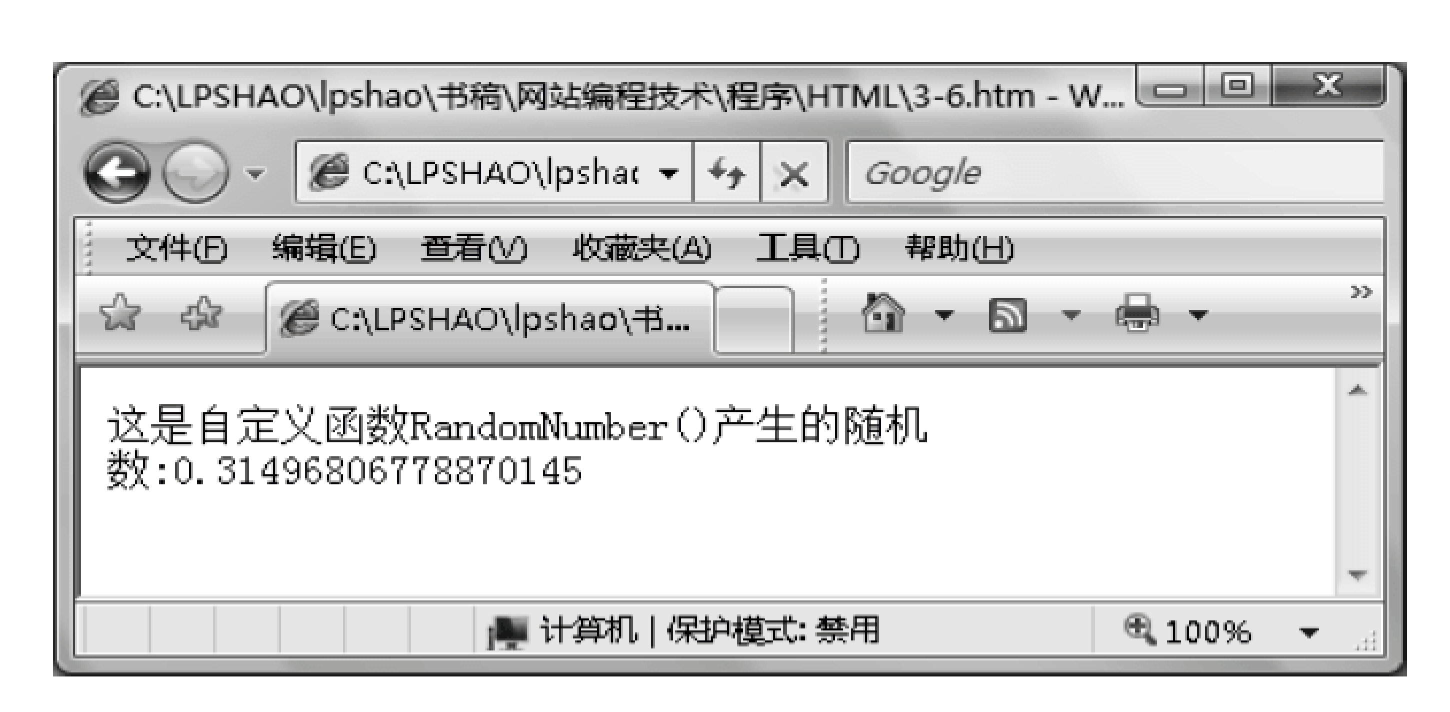


图 3.7 网页 3-6.htm 的运行结果

3. 知识点归纳

1) 创建自定义函数的方法

创建自定义函数的语法格式:

```
function 函数名(参数 1, 参数 2, …) {
        命令块
[return x]
}
```

功能：定义一个名称为"函数名"的函数，例如 RandomNumber()。通过函数名可反复调用该函数，调用函数将执行函数命令块中定义的所有任务，而且可以得到函数的返回值。

2）函数在 HTML 文档中的位置

自定义函数放在任何位置都可以，通常放在 HTML 文档的<head>头部，在主体中通过相关语句调用自定义函数。

3）函数名的选取规则

选取函数名与选取变量名的规则相同。

4）函数的参数

函数的参数要放在括号内，例如 f(x)，如果有多个参数，它们之间要用逗号隔开。参数是函数或方法用于其计算和处理的形式数据。参数放在函数名或方法名后面的圆括号对内，调用函数或方法时将使用具体的数据传递给参数，函数通过参数接收数据，进行数据处理和计算。需要说明的是函数也可以不带参数，例如 f()。

5）带返回值的函数

JavaScript 通过函数命令块中的 return 关键字给出函数的返回值，例如 return num。这样，调用随机函数 RandomNumber 时，即可得到 num 的值。JavaScript 函数也可以不带返回值，不使用 return 关键字即可。

6）函数的执行

函数不会自动运行，只有通过调用函数的命令才能运行该函数，执行函数中定义的功能。

3.4 JavaScript 的选择语句

本节主要解决如何有选择的执行 JavaScript 的命令语句等问题。

3.4.1 if-else 单选语句

1. 单选语句的作用

到目前为止，使用的 JavaScript 的命令语句都是按行的顺序逐行执行每条命令的。能否改变程序中 JavaScript 命令语句的执行顺序呢？

使用 if-else 单选语句可以根据条件选择执行某些命令语句，并跳过一些命令语句不执行。

2. 在程序中使用单选语句

例 3.7 一个使用 if-else 单选语句，根据不同的时间显示不同问候的文档(3-7.htm)。

```
<html>
<script language="JavaScript">
var mess1="";
document.write("<center><font color='#0000FF' size=4><b>");
day=new Date();
hr=day.getHours();
if ((hr>=6) && (hr<18))
  {mess1="现在是白天,该工作了。"};
else
  {mess1="现在是黑夜,该休息了!"};
document.write(mess1);
document.write("</b></font></center>");
</script>
</html>
```

在浏览器打开3-7.htm网页,可以看到如图3.8所示页面。

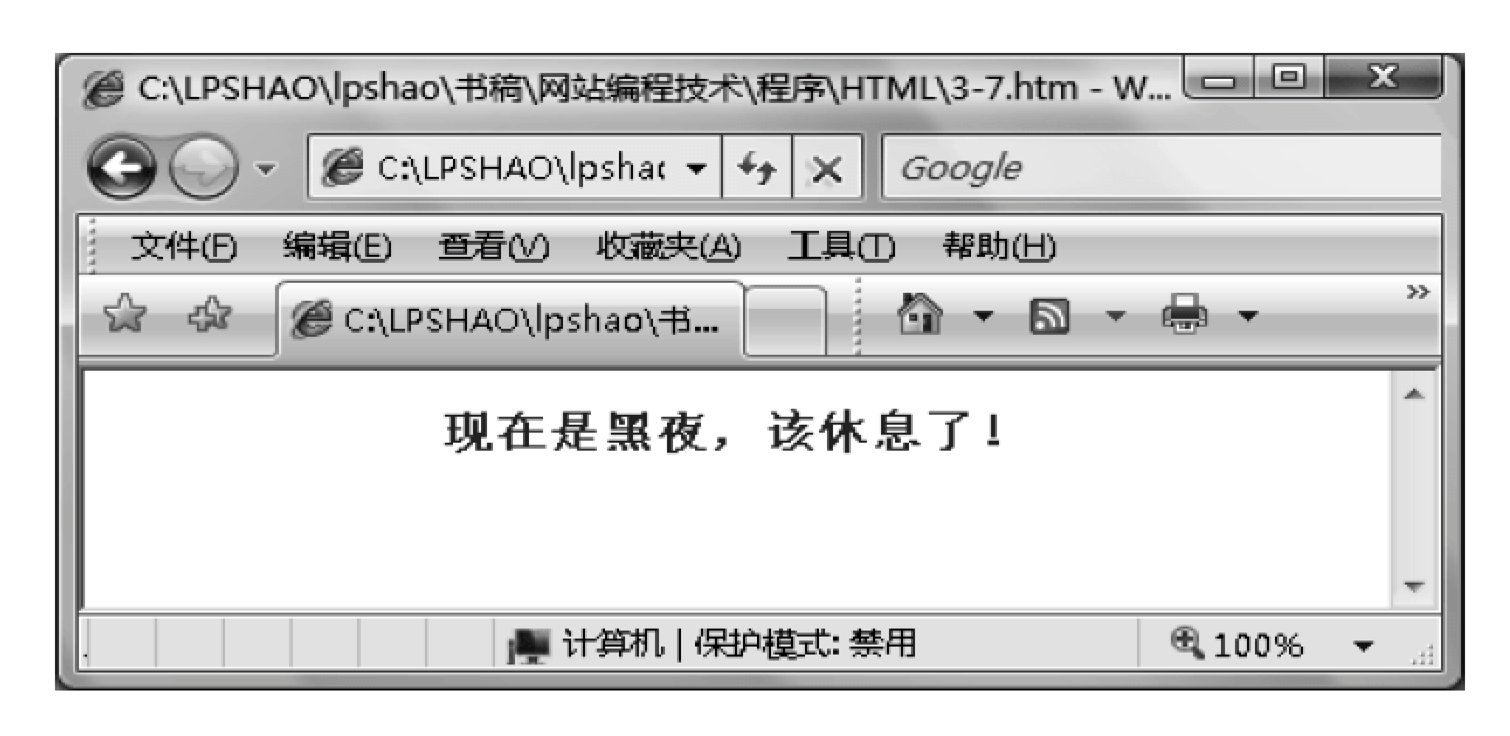


图3.8 网页3-7.htm的运行结果

3. 知识点归纳

1）if-else单选语句的结构

if-else语句的语法格式：

```
if 条件
    {JavaScript 命令}
else
    {其他命令}
```

功能：条件为真时执行if条件下面的命令,条件为假时执行else下面的命令。

2）嵌套使用的if-else语句结构

嵌套使用if-else的语法格式：

```
if 条件 1
    {JavaScript 命令}
else
```

```
{if 条件 2
    {JavaScript 命令}
else
{其他命令}
}
```

一定要配套使用。

3.4.2 switch 多选语句

1. switch 多选语句的作用

使用 if-else 单选语句可以解决两种情况下的选择问题。当遇到多种情况的选择问题时，虽然可以通过 if-else 的嵌套结构来解决，但程序代码太啰嗦，有没有其他的解决方案呢？

switch 多选语句是解决多种情况选择的最好方案。

2. 在程序中使用 switch 多选语句

例 3.8 使用 switch 多选语句，根据不同的时间显示不同问候的程序文档(3-8.htm)。

```
<html>
<script language="JavaScript">
var mess1="";
day=new Date();
hr=day.getHours();
switch ((hr>=0) && (hr<=4)?1:(hr>=5) && (hr<7)?2:(hr>=7) && (hr<12)?3:(hr>=12) && (hr<15)?
4:(hr>=15) && (hr<17)?5:(hr>=17) && (hr<18)?6:(hr>=18) && (hr<19)?7:8)
{
case 1:{document.write("深夜了,注意身体,该休息了!"); break;}
case 2:{document.write("清晨好,这么早就上网呀?"); break;}
case 3:{document.write("早上好,一天之际在于晨,又是美好的一天!"); break;}
case 4:{mess1="该吃午饭啦!有什么好吃的?"; break;}
case 5:mess1="又要开始工作了吧?有什么收获呀?";break;
case 6:mess1="该休息一下了!准备吃饭吧!";break;
case 7:mess1="吃过晚饭了吗?";break;
default:mess1="一天过的可真快!今天过的好吗?" ;
}
document.write(mess1)
</script>
</html>
```

在浏览器打开 3-8.htm 网页，可以看到如图 3.9 所示页面。

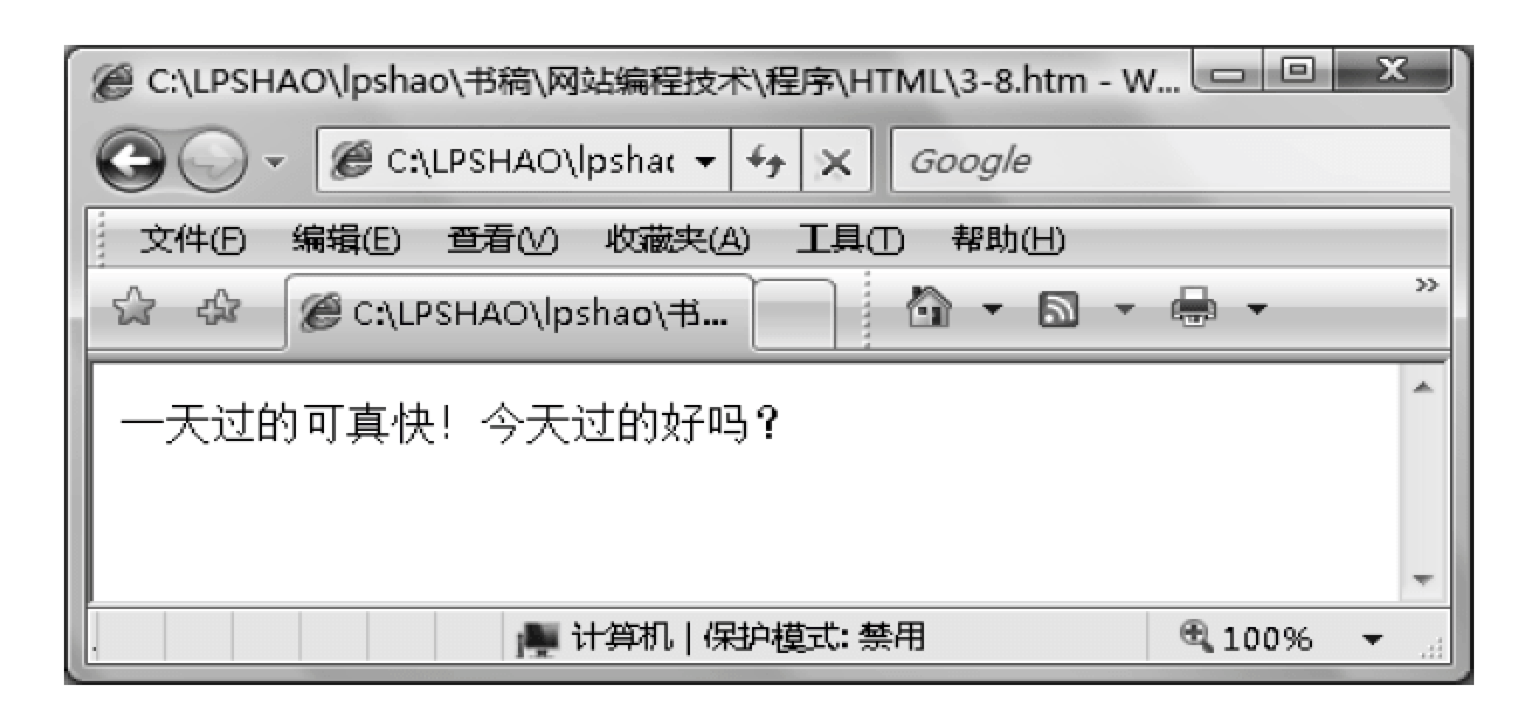


图 3.9　网页 3-8.htm 的运行结果

3. 知识点归纳

1）switch 多选语句的结构

switch 多选语句的语法格式：

```
switch (条件表达式) {
  case 常量值 1:语句 1; break;
  case 常量值 2:语句 2; break;
  ⋮
  default:语句 n;
}
```

2）switch 多选语句的执行顺序

先计算 switch 后的“条件表达式”，然后将表达式的值与各个 case 后的“常量值”比较，如果表达式的值与某个常量的值相等，就执行该常量值后面的语句。如果都不相等，就执行 default 下面的语句。碰到 break 语句程序流程将跳出 switch 语句体。

3）default 子句

default 子句可以没有命令语句，即什么都不执行，直接跳出分支语句。如果有命令语句，当 case 语句都不成立时，执行 default 子句中的命令。

4）使用 switch 多选语句要注意的事项：

- case 后面的常量必须是整数或字符型数据，而且不能有相同的值。
- 通常在每一个 case 中都应使用 break 语句提供一个出口，使程序流程能够跳出多选语句。否则，在第一个满足条件 case 后面的所有语句都会被执行，这种情况叫做落空。

3.5　JavaScript 的循环语句

本节主要解决如何重复执行某些 JavaScript 的命令语句等问题。

3.5.1 for 循环语句

1. 循环语句的作用

前面介绍的都是线性的程序，即每行程序命令只有一次执行的机会(选择语句可以跳过一些行代码不执行)。怎样重复执行一些命令语句呢？

循环语句可以解决这个重复执行一些命令语句的问题。

2. 在程序中使用 for 循环语句

例 3.9 使用 for 循环语句的程序文档(3-9.htm)。

```
<html>
<body>
<script language="JavaScript">
var name=prompt("请写出你的姓名：","lp");
document.write("<h5>这是"+name+"喜欢的 3 种活动</h5>");
for (var i=1;i<=3;i++){
document.write("<br>")
document.write(i+"."+prompt('活动'+i,'活动类型'));
}
</script>
</body>
</html>
```

在浏览器打开 3-9.htm 网页，输入三种喜欢的活动名称，可以看到如图 3.10 所示页面。



图 3.10 网页 3-9.htm 的运行结果

3. 知识点归纳

for 循环语句的语法格式：

```
for(初值;条件;步长表达式){
```

```
命令
}
```

功能：循环前先计算条件，值为 true(真)时，执行循环语句中的命令，值为 false(假)时跳出循环语句，每循环一次计数变量增加或减少一个步长单位的值。

其中，初值用来设置计数变量并给它赋值。初值表达式能够用 var 来声明一个新变量。步长表达式用来说明计数变量的变化规律。一般用 i、j、k、l 等作为循环的计数变量。

i++表示每循环一次 i 的值加 1。

3.5.2　for…in 循环语句

for…in 循环语句可用于在对象的所有属性内自动移位。

for…in 循环语句的语法格式：

```
for (j  in  对象) {
  命令语句
}
```

循环会从 0 增加 j，直到达到该对象中的最后一个属性的下标为止。

3.5.3　while 循环语句

while 循环语句的语法格式：

```
while (条件){
    命令
    break
    continue
}
```

循环前先计算条件表达式，值为真 true 时，执行循环语句中的命令，值为 false 时跳出循环语句。

while 循环中的条件可以是 JavaScript 逻辑表达式。条件为 true 是，执行循环体中的命令语句。break 语句可以随时终止循环，即使循环没有结束也如此。continue 语句可跳到循环体中命令语句的第一句开始下一次循环。

while 循环一定要注意其结束方式，以防陷入死循环中出不来。

break、continue 命令在其他循环中同样可用。

3.5.4　使用循环语句随机显示不同的图片

能否在浏览器打开页面时随机显示不同的图像文件呢？使用循环语句与随机函数可以解决这个问题。交通实现方案参考例 3.10。

例 3.10　使用 for 循环，随机显示不同图像文件的文档(3-10.htm)。

```
<html>
<script language="JavaScript">
```

```
a=3; var pp=Math.random(); var p1=Math.round(pp* (a-1))+1;
function c1() {
          this.src='';
          this.border='';
          this.alt='';  }
b=new Array()
for(var i=1; i<=a; i++) {b[i]=new c1()}
b[1].src="1.gif";b[1].border="0";b[1].alt="";
b[2].src="2.gif";b[2].border="0";b[2].alt="";
b[3].src="3.gif";b[3].border="0";b[3].alt="";
var pic="";
pic+='<img src='+b[p1].src+' border="0"'+' alt='+b[p1].alt+'>';
document.write(pic);
</script>
刷新会显示不同的图片
</html>
```

在浏览器打开 3-10.htm 网页,可以看到如图 3.11 所示的页面。单击"刷新"按钮,可以看到其他图像。

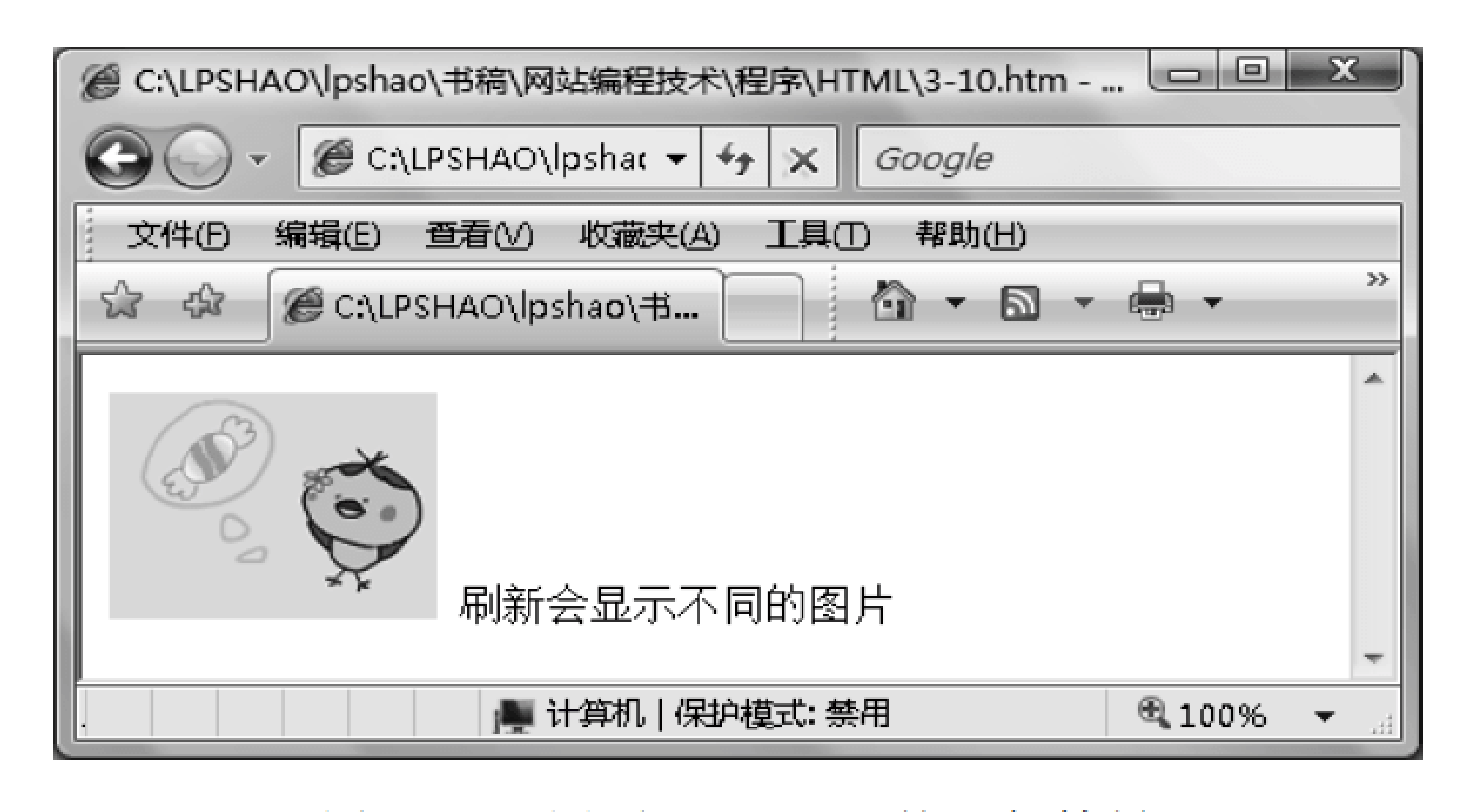


图 3.11 网页 3-10.htm 的运行结果

知识点归纳如下:

(1) 隐式方式声明变量的方式。在 3-10.htm 可以发现,声明变量时有的变量用了 var 关键字,有的没有用 var 关键字。不用 var 声明的变量是以隐式方式定义的变量。

(2) this 的作用。在 JavaScript 中 this 有特殊的作用,它表示当前的对象。这里的对象是指函数 c1。

(3) Array 数组对象。在 3-10.htm 还使用了 Array 数组对象。其实例对象 b 为一个一维的包含 3 个元素的数组变量。

(4) 使用图像文件夹。脚本程序用到的 3 张图片文件要先存放在与当前文件夹的子文件夹 img 中,名字分别为 1. gif、2. gif、3. gif。使用图像文件夹可以区别管理不同格式的文件。

3.6 JavaScript 的事件与事件处理程序

3.6.1 什么是 JavaScript 事件

1. 什么是事件

事件通常是指用户在页面上通过鼠标进行活动时产生的一种行为，如单击某个按钮就是一个事件，改变了表单的文本域或在链接上移动鼠标也是事件。事件不仅可以在用户交互过程中产生，浏览器自身的一些动作也可能产生事件。例如，打开浏览器载入一个网页时，就会产生一个 Load 事件。关闭一个页面时，会发生 unload 事件等。

归纳起来，JavaScript 的事件可以分为如下 3 种类型：

（1）页面之间跳转的事件，主要是超链接事件。

（2）浏览器自身引起的事件。

（3）表单内部元素同页面中元素的交互事件。

2. 事件处理机制

为了响应与处理事件，JavaScript 定义了常见事件的名称，JavaScript 常见的事件有 blur、click、change、focus、load、unload、mouseover、select、submit。事件名前加 on，称为事件处理器，例如 onChange 和 onClick 等。事件发生后浏览器将自动提供事件处理器信息，JavaScript 系统能够根据事件处理器名称，调用相应的事件处理程序对事件进行处理。这就是 JavaScript 的事件处理机制。

响应某个事件而进行的具体操作活动的程序称为事件处理程序。事件处理程序通过事件处理器来调用。事件处理程序可以是简单的 JavaScript 命令语句，也可以是自定义函数或对象实例方法，用来完成响应事件需要进行执行的任务与功能。表 3.1 是 JavaScript 中常用的一些事件及事件处理器名称。

表 3.1 JavaScript 事件表

事　件	应用对象	发生条件	事件处理器
Blur	窗口和所有表单元素	用户从窗口或表单中移开输入焦点	onBlur
Change	文本框、文本区、单选框等	用户改变元素值	onChange
Click	按钮、单选按钮、复选框、链接	用户在表单元素或在链接上单击	onClick
Error	图像、窗口	装载文档或图像时出错	onError
Focus	窗口和所有表单元素	用户在窗口或表单元素上聚焦	onFocus
Load	文档体	用户装载页	onLoad
MouseDown	文档、按钮、链接	用户按下一个鼠标键	onMouseDown
MouseMove	默认	用户移动光标	onMouseMove

续表

事　　件	应 用 对 象	发 生 条 件	事件处理器
MouseOver	链接	用户在链接上移动光标	onMouseOver
Move	窗口	用户或脚本移动窗口	onMove
Reset	表单	用户重置表单	onReset
Resize	窗口	用户或脚本改变窗口大小	onResize
Select	文本框、文本区	用户选择表单元素的输入域	onSelect
Submit	表单	用户提交表单	onSubmit
Unload	文档体	用户退出页面	onUnload

3.6.2　自定义函数构成的事件处理程序

在打开某个网站的页面时会出现是否将其添加到用户收藏夹的对话框窗口，这个功能是如何实现的呢？解决这个问题可以通过 onload 事件处理器及自定义函数构成的事件处理程序来解决。

1. 在程序中通过事件处理器调用函数

例 3.11　使用 onload 事件处理程序将页面添加到用户收藏夹的文档(为 3-11. htm)。

```
<html>
<head>
<script language="javascript">
function folder(){
window.external.AddFavorite('http://www.sina.com', '新浪首页')
}
</script>
</head>
<body onload=folder()>
你希望把新浪首页添加到收藏夹吗?
</body>
</html>
```

在浏览器打开 3-11. htm 网页，将弹出如图 3.12 所示对话框，单击“确定”按钮，可添加“新浪首页”地址到当前用户浏览器的收藏夹中。

2. 知识点归纳

1）load(载入页面)事件与 onload 事件处理器

3-11. htm 中在 load(载入页面)事件发生时，将触发 onload 事件处理器调用自定义函数 folder()，执行函数中定义的 JavaScript 命令完成指定的任务。



图 3.12　网页 3-11.htm 的运行结果

2）external.AddFavorite()方法

external.AddFavorite()方法的功能是在 IE 浏览器中自动打开“添加到收藏夹”对话框，在名称文本框会出现方法中确定的名称，如本例中的“新浪首页”，方法中的网址 http://www.sina.com 可以添加到收藏夹中。external.AddFavorite()方法只能在 IE 浏览器中使用。

3.6.3　由 JavaScript 命令构成的事件处理程序

对用户在页面文本框中输入文字或数据能进行验证吗？使用 JavaScript 的 onchange 事件处理器及事件处理程序即可解决这个问题。

1. 在程序中通过事件处理器调用 JavaScript 的命令语句

例 3.12　使用 onchange 事件处理器确认用户在页面中输入的数据的文档(3-12.htm)。

```
<html>
<body>
<form method=post>
请输入你的姓名：<br>
<input type="text" onchange="alert('感谢你的合作！');confirm('你想继续吗？')">
<p>
请输入一个小于 5 的数：<br>
<input type="text" onchange="if(parseInt(this.value)<=5){alert('请你输入一个比 5 大的
数')}">
<p>
</form>
</body>
</html>
```

在浏览器打开 3-12.htm 网页，可以看到如图 3.13 所示页面。

2. 知识点归纳

1）确认对话框

3-12.htm 中当 change(正在进行的活动改变时)事件发生时,触发 onchange 事件处理器直接调用 JavaScript 命令 alert('感谢你的合作!')与 confirm('你想继续吗?')打开提示对话框与确认对话框。

其中,confirm 方法用来显示一个带有"确定"和"取消"按钮的确认对话框并显示确认信息,单击"确定"按钮可继续输入,单击"取消"按钮可以开始其他操作,如图 3.14 所示。



图 3.13　网页 3-12.htm 的运行结果



图 3.14　确认对话框

2）转换函数 parseInt

3-12.htm 中使用转换函数 parseInt(this.value),将 string 类型的数据 this.value(用户在文本框输入的数据)转换为整型数据。

3.6.4　通过命令按钮实现数据计算功能

通过单击页面上的命令按钮,可以执行一个指定的事件处理程序。

1. 通过程序中的命令按钮调用事件处理器 onclick

例 3.13　通过"计算"按钮触发 onclick 事件处理器调用 c1(form)函数根据用户的输入进行数据计算,并给出计算结果的文档(3-13.htm)。

```
<html>
<script language="JavaScript">
  function c1(form){
    form.results.value=eval(form.entry.value);
  }
</script>
<body>
<form>
请输入一个算术表达式:<br>
```

```
<input type="text" name="entry" value=""><p>
这个表达式的结果为：<br>
<input type="text" name="results"><p>
<input type="button" value="计算" onclick="c1(this.form);">
</form>
</body>
</html>
```

在浏览器打开 3-13.htm 网页，可以看到如图 3.15 所示的页面。

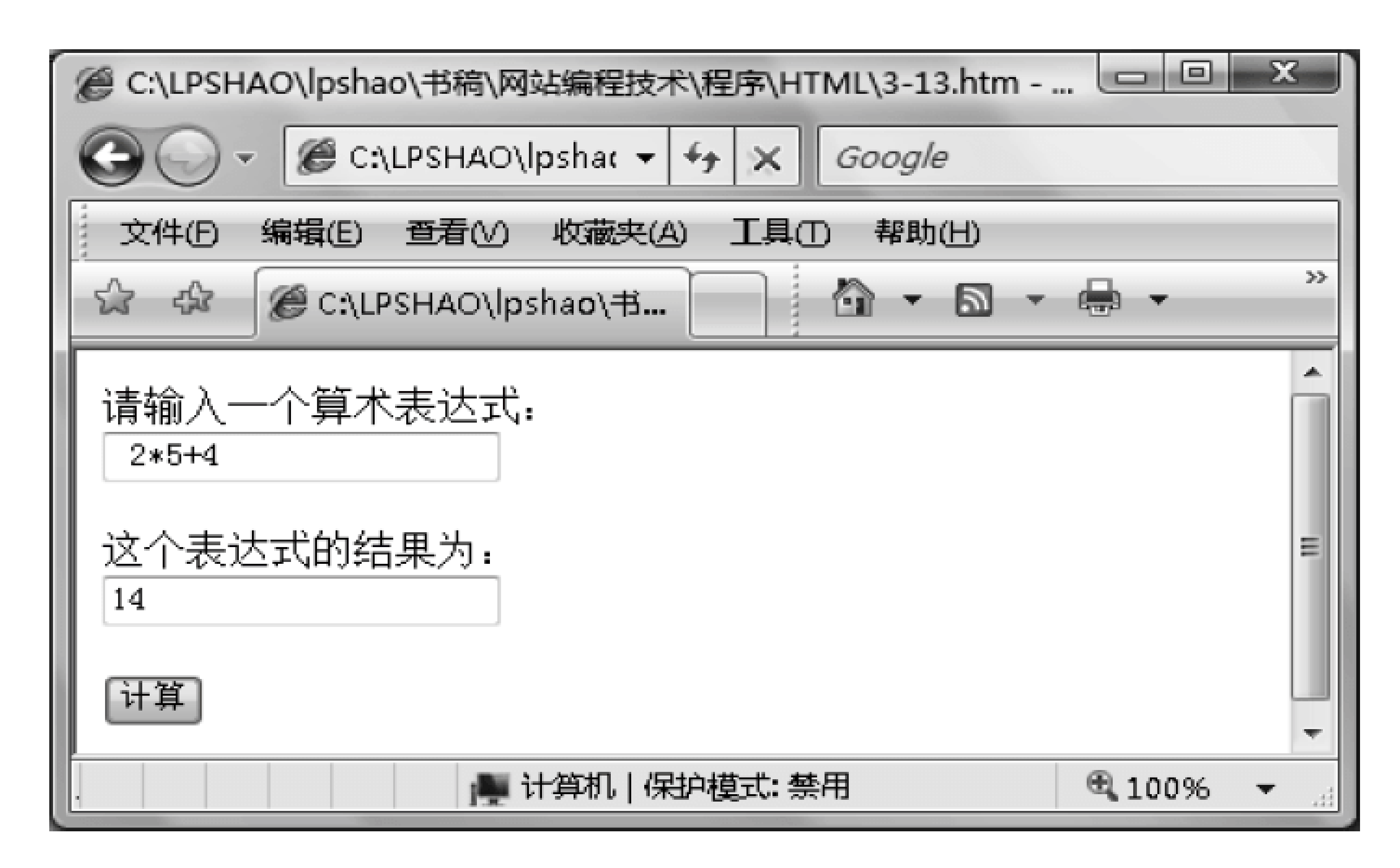


图 3.15　网页 3-13.htm 的运行结果

2. 知识点归纳

1）click 事件与 onclick 事件处理器

click 为单击按钮和超链接的操作行为事件。在 3-13.htm 中，在文本框输入一个算术表达式，如 2＊5＋4，然后单击“计算”按钮，将触发 onclick 事件处理器调用函数 c1，c1 将表达式的值 14 在文本框显示其结果。

2）独立函数 eval(表达式)

在 3-13.htm 中使用了独立函数 eval(表达式)，它的功能是计算表达式，返回表达式的值。

3.7　JavaScript 应用实例

3.7.1　创建保存 JavaScript 的脚本代码的 JS 文件

1. JS 文件的作用

能否将 JavaScript 的脚本代码与 HTML 代码分开存放在不同的文件中呢？这样可以将创建页面动态功能的任务与创建页面格式的任务分别保存在不同的文件中。

可以，JavaScript 的脚本代码可以单独存放在一个以.js 作为后缀名的文本格式文件

中。在 HTML 文档中通过指定.js 文件的存放路径,可以将.js 文件中编写的 JavaScript 的脚本代码载入到 HTML 文档中。

2. 创建 JS 文件

例 3.14 一个只包含 JavaScript 脚本程序代码的 JS 文本文件,脚本程序的任务是对输入的口令进行验证,文件名为 3-14.js。

```
var a,passwort;
a="1234";
passwort=prompt("请你输入口令!","");
if(passwort==a) location.href="3-15.htm";
else {
  alert("对不起,你输入的口令不对!");
location.href="3-16.htm";
}
```

保存文件时注意选择“保存类型”为“JScript 文件”。其文件后缀名会自动取为.js。如果使用其他网页编辑器,要注意保存文件为文本文件格式,并选取后缀名为.js。

3. 创建载入 JS 文件的 HTML 文档

例 3.15 包含 3-14.js 文件的 HTML 文档(3-14.htm)。

```
<html>
<head>
<script src="3-14.js" language="JavaScript" type="text/javascript">
</script>
</head>
</html>
```

4. 显示欢迎的 HTML 文件

例 3.16 显示欢迎文字的静态页面文档(3-15.htm)。

```
<html>
欢迎你进入我的网站!
</html>
```

5. 显示再见的 HTML 文件

例 3.17 显示再见文字的静态页面文档(3-16.htm)。

```
<html>
再见!
</html>
```

在浏览器打开 3-14.htm 网页,先看到如图 3.16 所示的对话框。如果在对话框中输

入的口令正确(1234),可打开3-15.htm欢迎页面,如图3.17所示。否则会在提示对话框中显示“对不起,你输入的口令不对!”,将打开再见页面。



图3.16 输入对话框

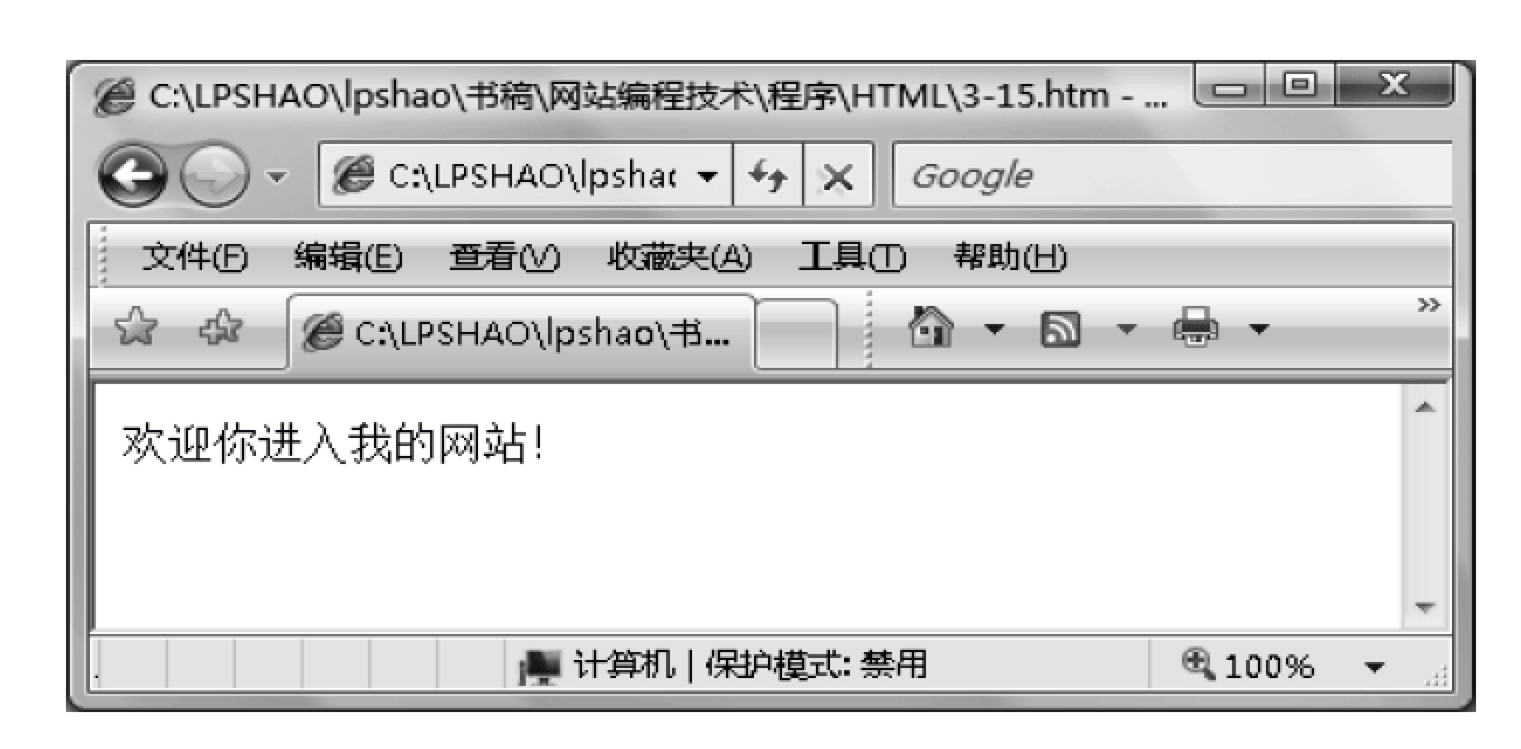


图3.17 网页3-14.htm在口令正确时打开的页面

6. 知识点归纳

1) .js文件

JavaScript代码可以单独存放在一个文件内,并以.js作为文本文件的后缀名。注意.js文件里不能有<script></script>或其他注释语句。

2) 在HTML文档调用JS文件的方式

在html文件中如果要使用.js文件,要在script标记码中使用src属性指定.js文件的存放路径。

参见下面的代码:

```
<script src="*.js" language="JavaScript" type="text/javascript"></script>
```

3.7.2 在状态栏显示滚动的文字

在网上的页面中你见到过在状态栏里滚动显示的文字吗?它们是如何实现的呢?

1. 创建具有动态功能的JS文件

例3.18 创建包含在状态栏上显示滚动文字函数的JS文件(3-17.js)。

```
var t1="我是滚动的文字,怎么样?很漂亮吧!你也试试吧。";
var lentxt=t1.length; var width=100; var pos=1-width;
function gd() {
```

```
    pos++; var scroller="";
    if (pos==lentxt) {pos=1-width;}
    if (pos<0) {
      for (var i=1; i<=Math.abs(pos); i++) {scroller=scroller+"";}
      scroller=scroller+t1.substring(0,width-i+1);
    }
    else {scroller=scroller+t1.substring(pos,width+pos);}
    window.status=scroller;
    setTimeout("gd()",100);
}
```

2. 创建包含 JS 文件的 HTML 文件

例 3.19 创建包含 3-17.js 文件的 htm 文件（3-18.htm）。

```
<html>
<script src="3-17.js" language="JavaScript" type="text/javascript">
</script>
<body onLoad="gd();">
注意!一会儿,你会看到在状态栏里出现滚动的文字
</body>
</html>
```

在浏览器打开 3-18.htm 网页，可以看到如图 3.18 所示页面。



图 3.18　网页 3-18.htm 的运行结果

3. 知识点归纳

1）setTimeout(…)定时器

在页面状态栏中可以看到滚动的文字，是因为使用了 setTimeout(…)定时器，setTimeout 方法指定时间 100 毫秒后，调用 gd()函数 scroller 变量向前移动一步。

2）具有多个知识点的 JS 文件

3-17.js 文件有点复杂，综合使用了函数、if-else 选择结构、for 循环结构等知识点，一下子不容易看明白。可以先模仿使用，只要把“我是滚动的文字，怎么样？很漂亮吧！你也试试吧。”换成你想显示的文字即可。然后，慢慢理解程序的编写方法。

3-18.htm 页面中状态栏里的文字会一直滚动，如果不想让它滚动下去，该如何做呢？请读者思考如何解决这个问题。

3.7.3 在页面上滚动显示的菜单

1. 创建 JS 文件

例 3.20 创建在页面上具有滚动菜单功能的.js 文件(3-19.js)。

```
var index=7
link=new Array(6); text=new Array(6);
link[0]='3-1.htm'; link[1] ='3-2.htm';link[2]='3-3.htm';link[3]='3-4.htm';
link[4]='3-5.htm'; link[5]='3-6.htm';link[6]='3-7.htm';
text[0]='新闻 1'; text[1]='新闻 2';
text[2]='菜单一'; text[3]='菜单二';
text[4]='菜单三'; text[5]='菜单四'; text[6]='菜单五';
document.write ("<marquee scrollamount='1' scrolldelay='100' direction='up' width='150'
height='150'>");
for (i=0; i<index; i++){
  document.write ("  <img src='img/1.gif' width='12' height='12'><a href="+link
[i]+" target='_blank'>");
  document.write (text[i]+"</a><br>");
}
document.write("</marquee>")
```

2. 创建包含 JS 文件的 HTML 文件

例 3.21 创建包含 3-19.js 文件的.htm 文件(3-20.htm)。

```
<html>
<script src="3-19.js" language="JavaScript" type="text/javascript">
</script>
    <body>
注意!一会儿,在页面上能看到滚动的菜单。
</body>
</html>
```

在浏览器打开 3-20.htm 网页，可以看到如图 3.19 所示页面。

3. 知识点归纳

1) marquee(滚动)标记码

菜单之所以能滚动，是因为使用了 marquee 滚动标记码，能向上滚动是在 direction(方向)属性中取值为 up。

2) document.write 方法的作用

一般使用 document.write 方法在页面上动态显示文本内容，在 write 方法参数中可

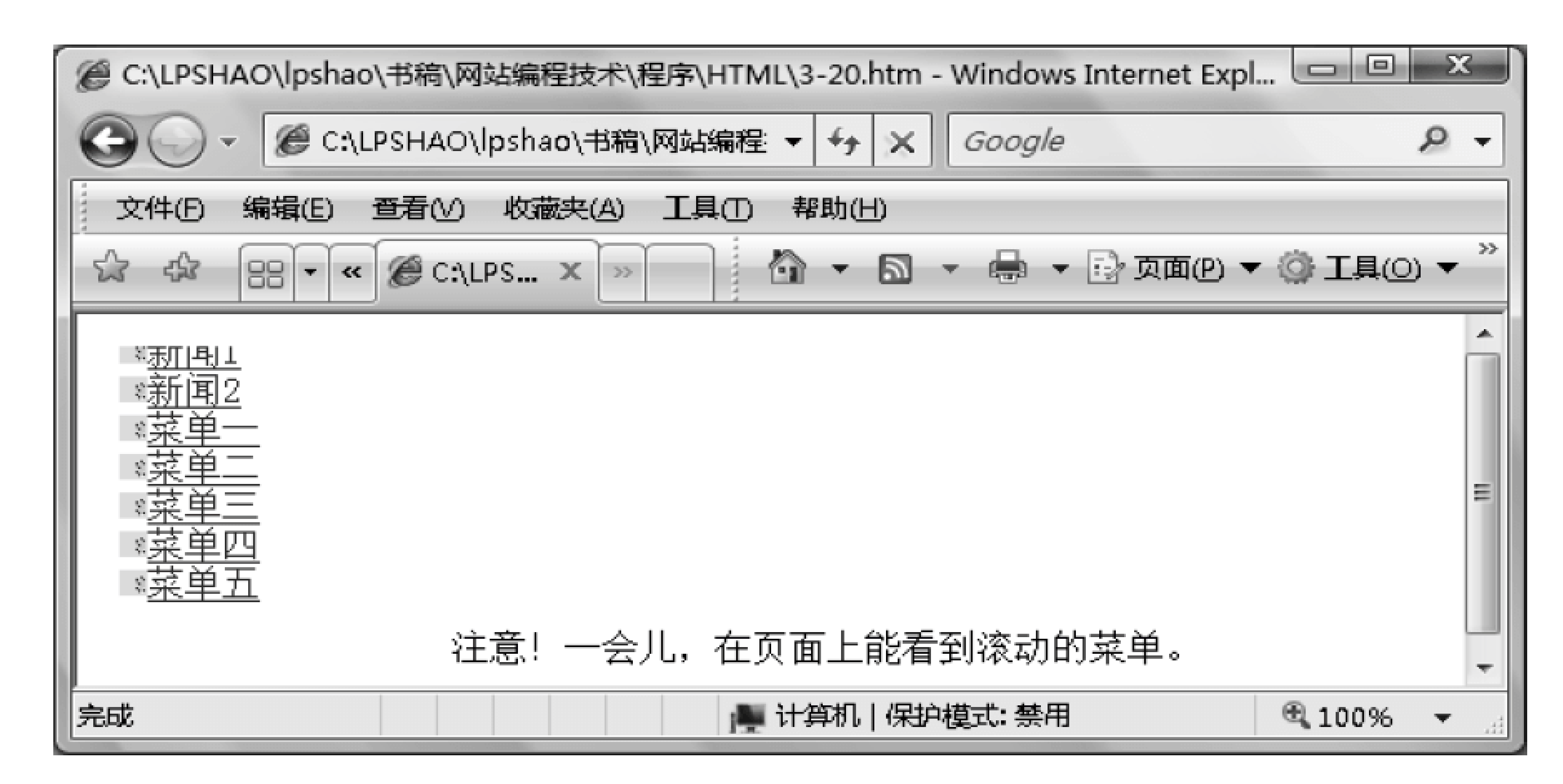


图 3.19　网页 3-20.htm 的运行结果

以嵌入 HTML 标记码，这样可以在页面上显示不同格式的文本内容。

3.8　思考与练习

3.8.1　思考题

3-1　JavaScript 有什么特点？

3-2　如何将 JavaScript 脚本代码嵌入到 HTML 文档？

3-3　JavaScript 有哪些常见对象？

3-4　如何定义 JavaScript 变量？ 变量与常量有什么区别？

3-5　如何创建 JavaScript 自定义函数？

3-6　JavaScript 有哪些类型的表达式？

3-7　JavaScript 有哪些类型的事件？ 如何触发事件过程？

3.8.2　上机练习

3-1　编写一个可以随意确定 N 的数值并显示出 $1+2+\cdots+N$ 的和的页面。

3-2　编写一个可以根据输入不同购物款数额，如 500 元以下、500～1000 元、1000～2000 元，分别显示给不同优惠券的页面。

3-3　编写一个包含计算圆面积函数 Y1，并可随意输入半径并显示圆面积值的页面文件。

3-4　编写一个包含计算立方函数 Y2，使用文本框输入数值，通过单击“确定”按钮调用 Y1 并显示该数值立方值的页面文件。

3-5　编写一个动态网页，实现在不同时间显示不同的问候、不同图片的功能。

3-6　编写一个页面文件：包含名称为“计算”按钮，当鼠标移到“计算”按钮上在状态栏显示“计算”功能的说明文字，鼠标离开“计算”按钮清除状态栏文字的功能。

3-7　编写一个带有滚动菜单的动态网页。



层叠样式表——CSS

CSS 是 Cascading Style Sheets(层叠样式表)的缩写,通常简称为样式表。顾名思义,它是一种设计样式的技术。

在制作页面时采用 CSS 技术,可以有效地设置页面的布局、字体、颜色、背景和其他效果,对页面实现更加精确的控制。描述样式表的 CSS 代码可以嵌入在 HTML 文档中,通过浏览器解释执行,CSS 语句的格式与 HTML 语句格式基本相同,非常容易使用。

本章主要解决以下问题:

- 在 HTML 文档中如何使用 CSS 样式;
- 如何通过 CSS 属性设置页面布局;
- 如何通过 CSS 定位属性指定页面元素在页面中显示的位置与层次;
- 如何通过 CSS 滤镜属性使元素对象显示特殊效果;
- 如何编写与使用单独保存的 CSS 文件。

4.1 CSS 的基本功能

本节简单说明 CSS 的基本功能。

使用 CSS 技术,只要对相应的代码做一些简单的修改,就可以改变同一页面的不同部分,或不同页面的网页外观和格式,使网页变得多彩多姿。

CSS 具有如下的基本功能:

- 在几乎所有的浏览器上都可以使用。
- 一些通过图片实现的功能,可用 CSS 实现,从而使页面可以更快地下载。
- 利用 CSS 属性可以设置字体、颜色、背景等页面格式,可使页面的字体变得更漂亮,更容易编排。
- 利用 CSS 定位可以使页面布局更加规范、美观,轻松地控制页面的布局。
- 可以统一多个网页的风格,使网页格式同时更新。可以将站点上所有的网页风格都使用一个 CSS 文件进行控制,只要修改这个 CSS 文件,那么整个站点的所有页面都会随之发生变动。
- 利用 CSS 滤镜属性可以使页面产生多媒体效果。

4.2 在 HTML 文档中使用 CSS 样式的方法

本节主要解决如何在 HTML 文档中使用 CSS 样式的问题,将给出 4 种不同的解决方案。

4.2.1 通过标记码定义 CSS 样式

如同在 HTML 文档中使用 JavaScript 语言一样,可以通过样式标记码<style>在 HTML 文档中引入 CSS 代码定义页面的样式。

1. 在 HTML 文档中引入 CSS 代码

例 4.1 通过样式标记码<style>引入 CSS 代码,在 CSS 代码中定义超链接和文字背景颜色的方式的 HTML 文档(4-1.htm)。

```
<html>
<head>
  <style>
    a {color:red}
    p {background-color:blue; color:white}
  </style>
</head>
<body>
  <a href="http://www.nease.net ">动态网页制作</a>
  <p>你注意到这一段文字的颜色和背景颜色了吗?</p>怎么样?
</body>
</html>
```

在浏览器打开 4-1.htm 网页,可以看到如图 4.1 所示页面。

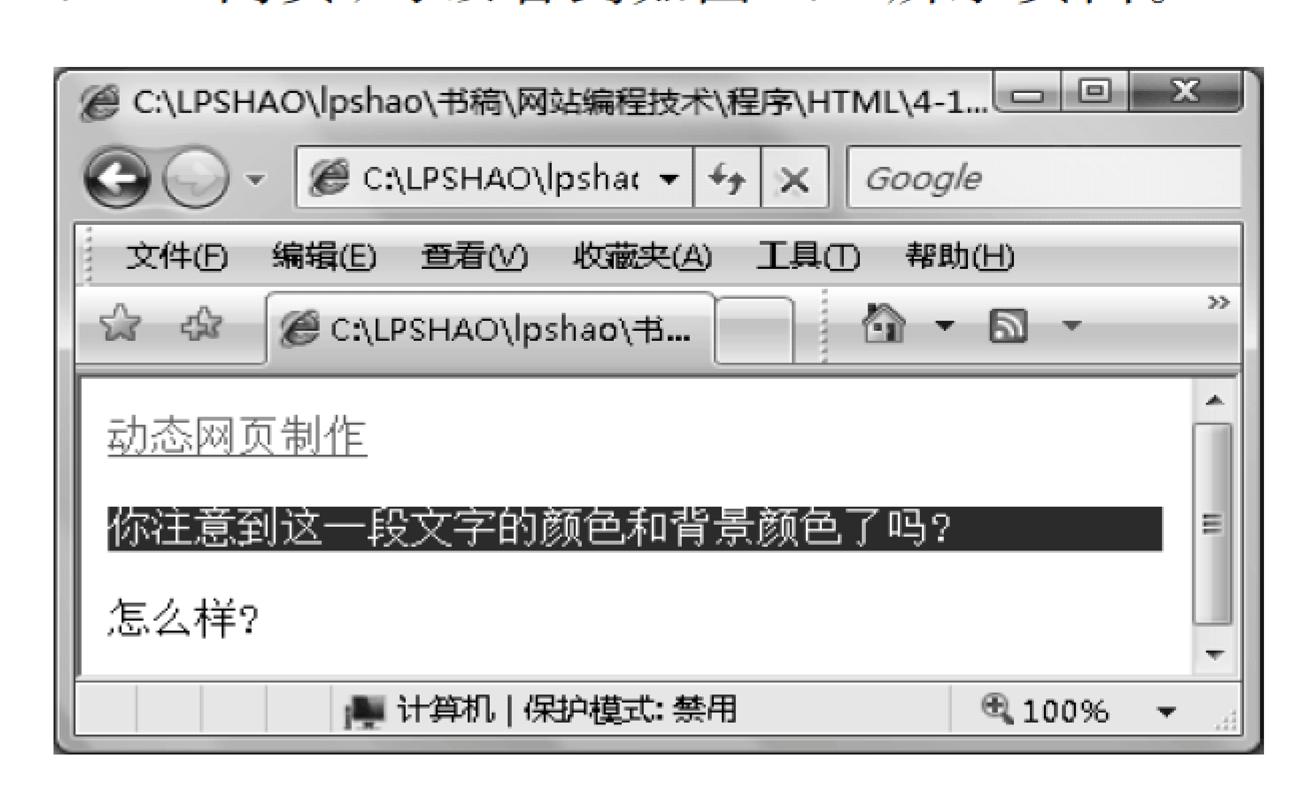


图 4.1 网页 4-1.htm 的显示结果

2. 知识点归纳

1) 在 HTML 文档中引入 CSS 代码的标记码

在 HTML 文档中引入 CSS 代码要使用<style></style>标记码。包含 CSS 代码

的<style></style>标记码对一般包含在<head>标记码内。

2）通过标记码统一定义样式

通过标记码统一定义 CSS 样式的语法格式：

```
标记码{属性名:属性值; 属性名:属性值; …}
```

其中，标记码为 HTML 的标记码，如 p、body、a 等。“属性名”指定哪种属性，如 color 为颜色属性。“属性值”用来设置属性的具体样式，如文本的颜色(color)属性的具体样式为红色(red)。这种定义样式的方式称为统一定义样式方式。

CSS 代码是由标记码、属性、属性值 3 部分组成的，它们通常用来定义 HTML 文档中显示元素的样式，使得标记码标识的元素可以按 CSS 代码定义的样式在页面中显示。

可见，CSS 代码是对 HTML 标记码的一个补充，可灵活定义标记码标识的元素的显示样式。

3）对多个标记码同时定义相同的样式

如果希望多个标记码标识的元素具有相同的样式，可参考下面的语句，例如：

```
h1, body, a {color:red}
```

这行 CSS 代码能让所有的 h1、body、a 标记码对中标识的文本元素颜色都为红色。

4.2.2 通过标记码的 style 属性定义 CSS 样式

可以通过其他方式插入 CSS 代码吗？有，可以直接在标记码中通过其 style 属性插入指定该标记码样式的 CSS 代码。

1. 通过标记码的 style 属性定义 CSS 样式的方式

例 4.2 通过标记码的 style 属性定义 CSS 样式的 HTML 文档：利用<body>标记码的 style 属性定义 CSS 样式，确定文本的字体大小、字体样式和文字颜色的方式(4-2.htm)。

```
<html>
  <body style="font-size:72;font-style:bold;color:red">你好</p>
  </body>
</html>
```

在浏览器打开 4-2.htm 网页，可以看到如图 4.2 所示页面。

2. 知识点归纳

1）直接定义方式

通过 HTML 标记码的 style 属性可以嵌入 CSS 代码定义样式，这种方式称为直接定义样式方式。在某个范围或单个标记中定义某种样式时可以使用这种方式。

2）直接定义样式的语法格式

通过标记码的 style 属性嵌入 CSS 定义代码的语法格式：



图 4.2　网页 4-2.htm 的显示结果

```
<标记码 style="属性名:属性值;属性名:属性值;属性名:属性值;…" >…</标记码>
```

CSS 代码由 style=“属性:属性值”组成，用来定义指定标记码的样式。

在例 4.2 中，font-size、font-style 和 color 为确定的属性名，分别指定的是字体大小(size)、字体样式(style)和文字颜色(color)属性；而 72、bold、red 为具体样式的属性值。CSS 样式代码的含义是以 72pt、粗体、红色的样式显示标记码标识的文本文字。

4.2.3　通过标记码的 Class 属性定义 CSS 样式

通过“统一定义”方式可以对 HTML 同一类型的标记码或多种类型的标记码定义同一种 CSS 样式。通过“直接定义”方式可以对 HTML 单个标记码定义一种 CSS 样式。

如果希望对同一类型的部分标记码设置相同的样式，部分标记码设置另外的样式该如何定义呢？可先定义相关类 Class，然后通过 HTML 标记码的 Class 属性解决这个问题。

1. 同一类型的标记码以不同样式显示

例 4.3　在页面中通过相关类及 class 属性为 H2 标记码设置不同颜色样式的文档(4-3.htm)。

```
<html>
<head>
<style>
  h2.s1{color:red}
  h2.s2{color:blue}
</style>
</head>
<h2>没有改变的 h2</h2>
<h2 class=s1>改变为红色的 h2</h2>
<h2 class=s2>改变为黄色的 h2</h2>
<h2 class=s1>第一章</h2>
<h2 class=s2>第二章</h2>
```

```
</html>
```

在浏览器打开 4-3.htm 网页，可以看到如图 4.3 所示页面。

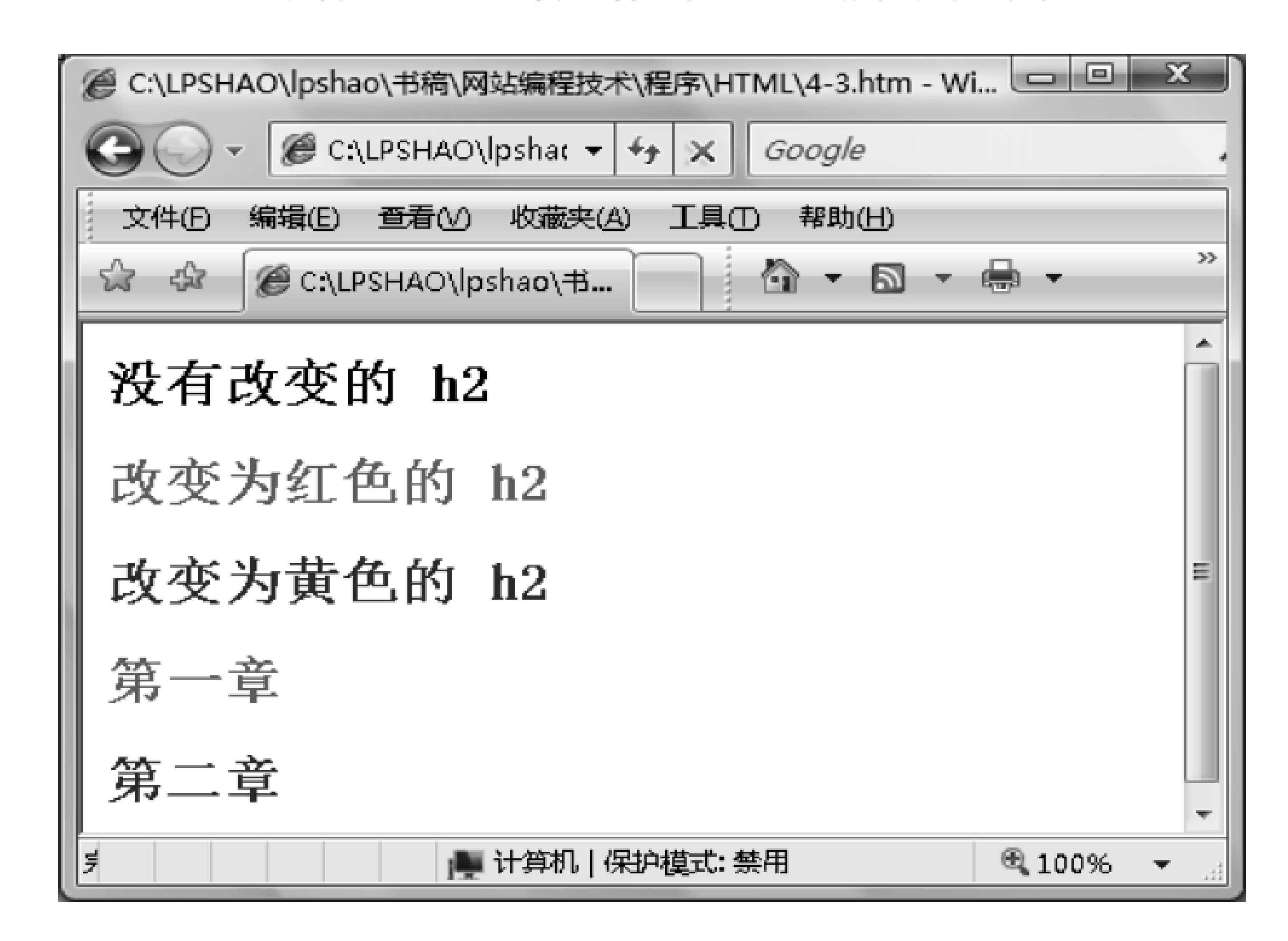


图 4.3　网页 4-3.htm 的显示结果

2. 不同类型的标记码以相同样式显示

例 4.4　在页面中通过独立类及 class 属性设置某些 H2、P 标识的文本颜色为蓝色的文档(4-4.htm)。

```
<html>
<head>
<style>
  .s2{color:blue}
</style>
</head>
<p class=s2>这是蓝色的 p</p>
<h2 class=s2>这是蓝色的 h2</h2>
<h2>这是黑色的 h2</h2>
</html>
```

在浏览器打开 4-4.htm 网页，可以看到如图 4.4 所示页面。

3. 知识点归纳

CSS 提供了相关类与独立类来解决多个标记码设置相同样式与相同标记码设置不同样式的问题。

1) 通过相关类定义样式

通过相关类定义样式中包含两部分内容：HTML 标记码与自定义类名、属性名与属性值。通过相关类定义的样式可以分别用在相同标记码中，其语法格式如下：

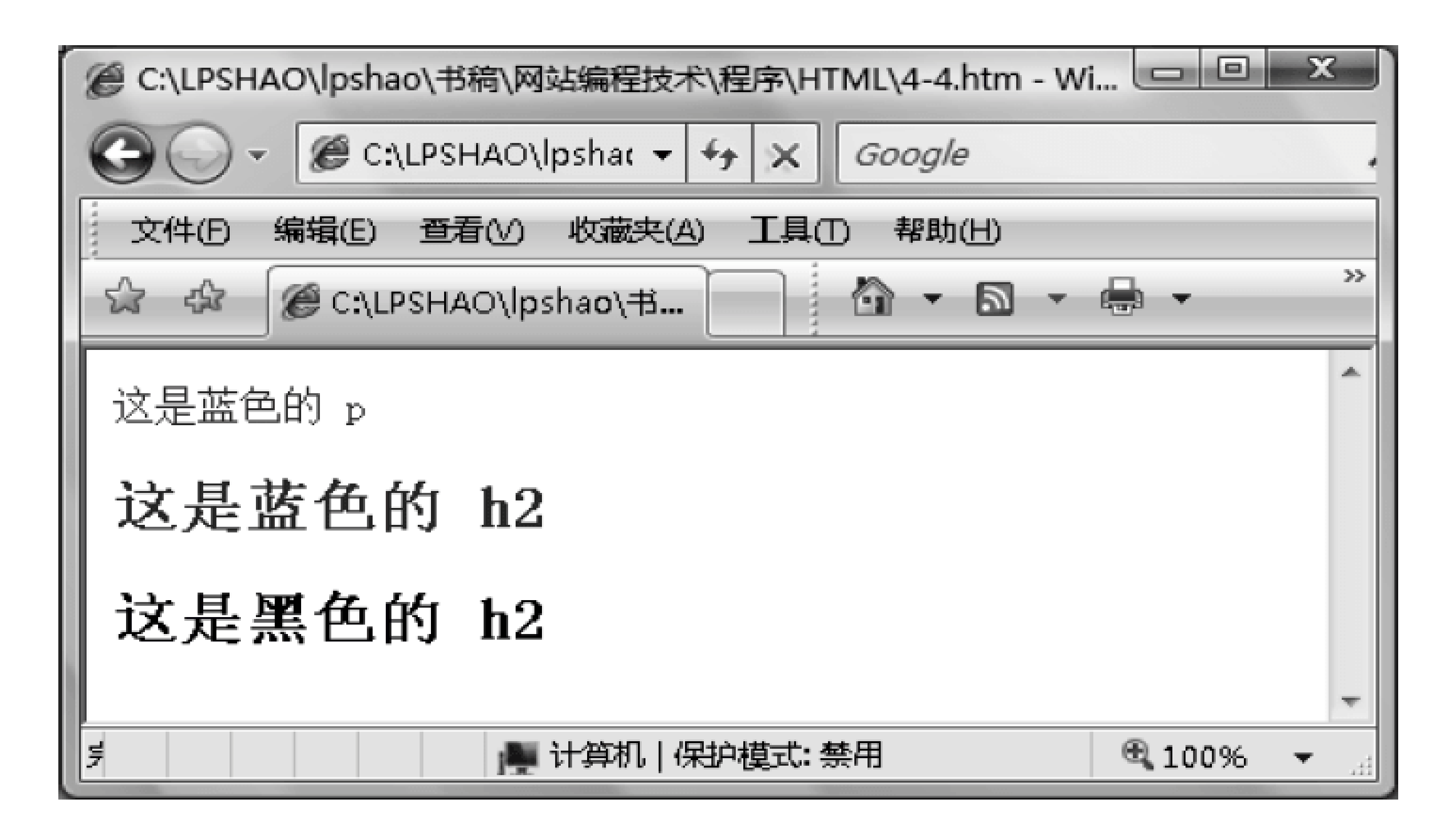


图 4.4　网页 4-4.htm 的显示结果

```
标记码.类名 {属性名:属性值}
```

2）通过独立类定义样式

通过独立类定义样式中包含两部分内容：自定义类名、属性名与属性值。通过独立类定义的样式，可用在 HTML 的多个不同的标记码上，其语法格式如下：

```
.类名 {属性名:属性值}
```

独立类样式能更加自如地对标记码定义 CSS 样式。

3）通过 Class 属性选择 CSS 样式

使用标记码的 Class 属性选择样式分为两个步骤：首先要通过<style>标记码引入 CSS 代码定义相关类或独立类定义的 CSS 样式；然后在 HTML 文档中通过指定标记码的 Class 属性选择类名，根据类名选择定义好的样式。

4.2.4　通过标记码的 ID 属性定义 CSS 样式

通过 ID 属性定义 CSS 样式与通过 class 属性定义 CSS 样式功能相同基本，区别在于它需要先定义称为 ID 名的 CSS 样式，然后通过标记码的 ID 属性选择 ID 名使用 CSS 样式。

1. 不同类型的标记码以相同样式显示

例 4.5　在页面中为 H3、P 标记码设置 ID 属性、显示不同颜色样式的文档(4-5.htm)。

```
<html>
<head>
<style>
  #s1 {font-style:bold; color:red;}
  #s2 {font-style:bold; color:blue;}
</style>
<P ID="s1">这是红色的 P</P>
<H3 ID="s1">这是红色的 H3</H3>
```

```
<H3 ID="s2">这是蓝色的 H3</H3>
</head>
</html>
```

在浏览器打开 4-5.htm 网页，可以看到如图 4.5 所示页面。

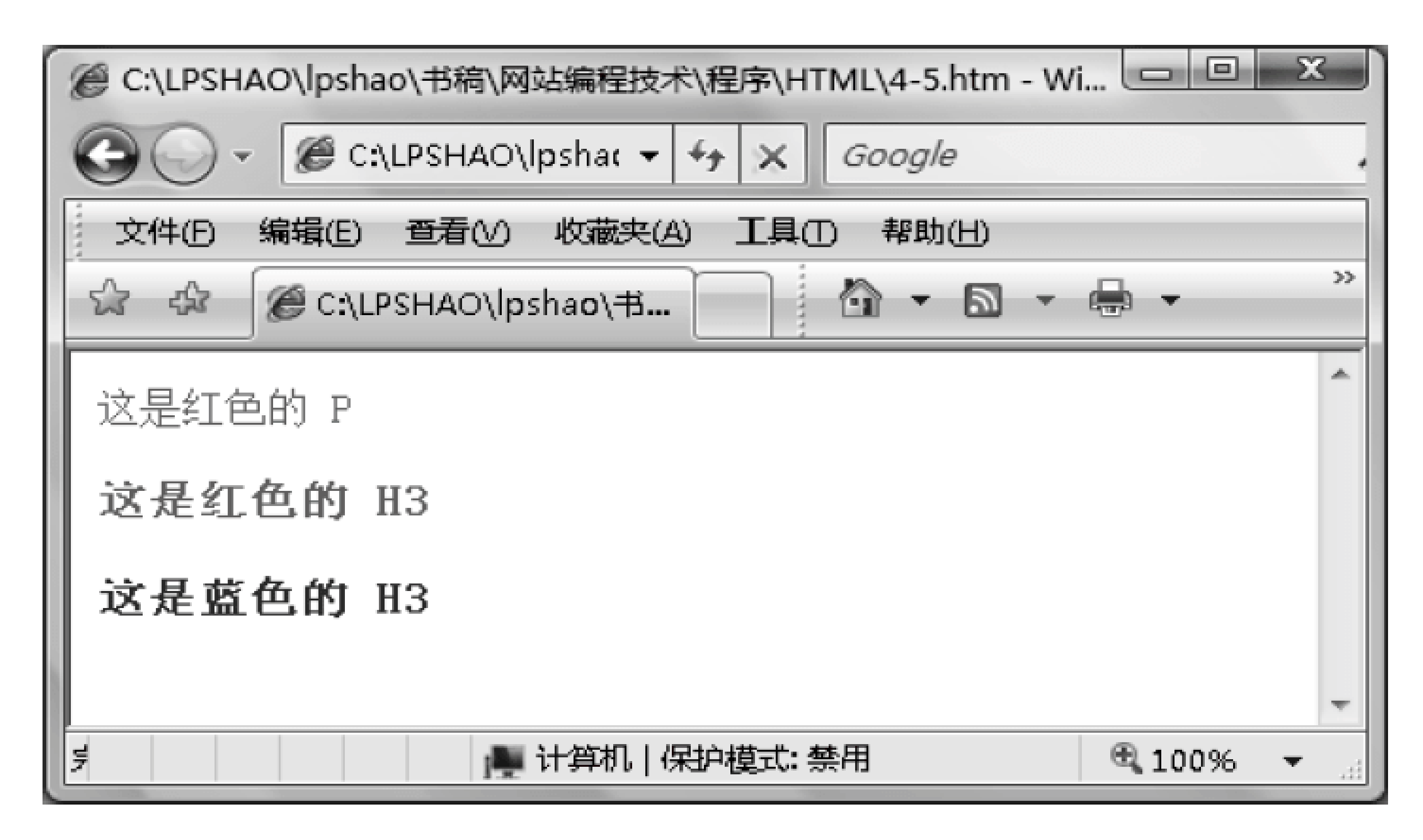


图 4.5　网页 4-5.htm 的显示结果

2. 知识点归纳

通过 ID 名定义 CSS 样式的语法格式：

```
#IDname {属性名:属性值}
```

既然通过 ID 属性与 class 属性定义 CSS 样式的功能基本相同，何必要设计两种方式呢？其实，通过 ID 属性定义 CSS 样式有特别的功能，后面的内容会介绍。

4.3　使用 CSS 属性设置页面布局

从上面介绍的 CSS 语法格式可以看出，属性是 CSS 非常重要的部分。熟练掌握 CSS 的各种属性，可以得心应手地编辑页面布局。

本节的内容主要解决如何通过 CSS 的属性设置或修改 HTML 文档中的字体、字间距、行间距、颜色、背景、边距、填充距、位置、可视性和空间位置等页面元素的布局问题。

4.3.1　网页中使用的单位

CSS 样式表的另一个优势在于它可以精确地定义和控制元素的外观。通过 CSS 样式可以设定显示元素在窗口页面中的宽、高、左右边距以及字体大小等。

要确定元素在页面中的大小、位置，需要先了解网页文件中使用的单位。网页文件中使用的单位一般分为相对单位和绝对单位两种。

1. 绝对单位

网页上使用的绝对单位有厘米(cm)、毫米(mm)、英寸(in)、点(pt)、派克(pc)。绝对

单位通常不能适应各种浏览器多样性的变化，因此应用的比较少。

2. 相对单位

网页上常用的单位是相对单位，相对单位是指元素尺寸相对于浏览器系统默认值相对的缩放。包括像素（px）、em 和 ex。px 是相对浏览者计算机屏幕分辨率来显示的，Windows 用户一般使用 96 像素/英寸的分辨率。em 是相对于当前对象中文本 M 的字体尺寸大小，如果没有设置当前对象文本的字体尺寸，则使用浏览器默认文本 M 的字体尺寸。ex 是相对于当前对象中文本 x 的字体尺寸大小。

4.3.2 字体属性

如何使用 CSS 代码设置页面中显示文本的字体属性呢？

1. 设置字体属性的方式

例 4.6 设置不同字体属性的文档（4-6.htm）。

```
<html>
<head>
<style>
  #s1 {font-style:italic;font-variant:small-caps;font-weight:bold;font-size:28pt;font-family:黑体}
  #s2 {font:italic small-caps bold 28pt 黑体}
</style>
</head>
<p ID="s1">字体属性的使用 abcABC</p>
<p ID="s2">字体属性的使用</p>
</html>
```

在浏览器打开 4-6.htm 网页，可以看到如图 4.6 所示页面。

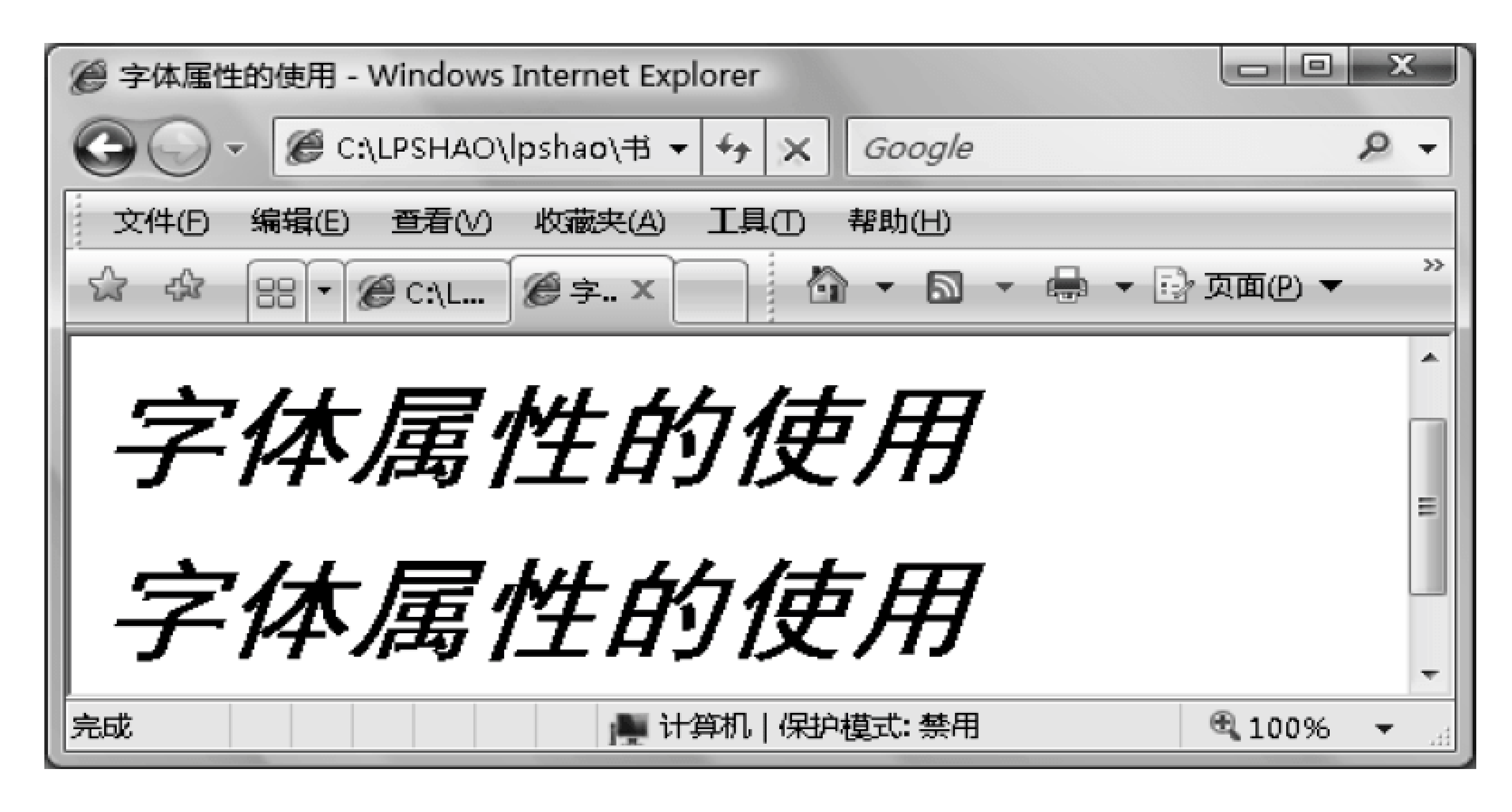


图 4.6 网页 4-6.htm 的显示结果

2. 知识点归纳

(1) 通过 font 可以设置字体的各种属性。

(2) 设置 font 的属性的简化方式。

设置 font 的属性可以使用完整方式,即说明"属性:属性值",还可以使用简化方式,只是说明 font 的属性值。例如:

```
{font:italic small- caps bold 28pt 黑体}
```

但要注意属性值的排放顺序。其顺序为 font-style、font-variant、font-weight、font-size、font-family,其中没有定义的属性值系统会以默认值显示。

(3) 表 4.1 是 CSS 关于字体的基本属性,在编辑页面时经常会用到。

表 4.1 字体属性

属　性	属性含义	属　性　值
font-family	使用什么字体	所有的字体
font-style	字体是否斜体	Normal、italic、oblique
font-variant	字体是否用小体大写	Normal、small-caps
font-weight	定义字体的粗细	Normal、bold、bolder、lithter 等
font-size	定义字体的大小	Absolute-size、relative-size、length、percentage 等

其中,small-caps 为小体大写字体,意为小字体大写字母。

4.3.3 颜色与背景属性

1. 通过 CSS 代码定义文字的颜色与背景属性

例 4.7 为页面文字设置颜色与图像背景的网页文件(4-7.htm)。

```
<html><head>
<style>
p{color:red;background- image:url (3.GIF);background- repeat:no- repeat; background-
position:top center}
</style></head>
<p style="font:bold">文字的前景色、背景图片、背景图片的位置</p>
</html>
```

在浏览器打开 4-7.htm 网页,可以看到如图 4.7 所示的页面。

2. 知识点归纳

(1) 通过 CSS 的颜色属性 color 可以设置对象的前景色,通过 CSS 的背景属性 background 可以设置对象的背景颜色、背景图片。

(2) CSS 的颜色和背景属性如表 4.2 所示。

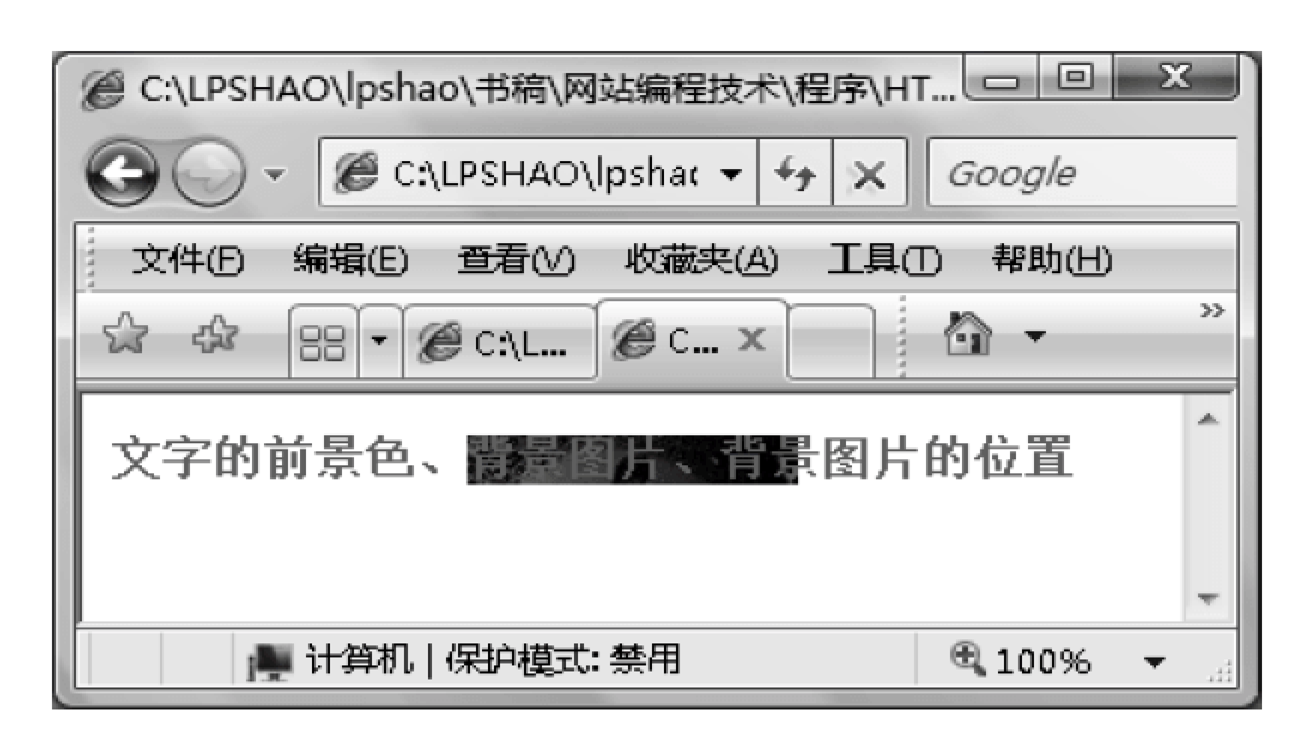


图 4.7 网页 4-7.htm 的显示结果

表 4.2 CSS 的颜色和背景属性

属　　性	属 性 含 义	属性书写格式	属　性　值
Color	定义前景色	例：p {color：red}	颜色
Background-color	定义背景色	例：body {background-color：yellow}	颜色
Background-image	定义背景图案	例：body {background-image：url(.jpg)}	图片路径
Background-repeat	背景图案重复方式	例：body {background-repeat：repeat-y}	repeat-x、repeat-y、norepeat
Background-attachment	设置滚动	例：body {background-attachment：scroll}	Scroll Fixed
Background-position	背景图案的初始位置	例：body {background：url(.jpg)top center}	Percentage、length、top、left、right、bottom 等

4.3.4 文本的排版属性

如何使用 CSS 代码对页面中的文本进行排版呢？

1. 任务与步骤

例 4.8 在网页文件中显示文本文字的文档(4-8.htm)。

```
<html>
<body>网海茫茫，怎样使自己的网页成为一道亮丽的风景线让人过目不忘？
怎样使自己的网页的风格独树一帜？怎样免除众多 HTML 标志属性设置的烦恼？
CSS---层叠样式单可以助你一臂之力！
</body>
</html>
```

在浏览器打开 4-8.htm 网页，可以看到如图 4.8 所示的页面。

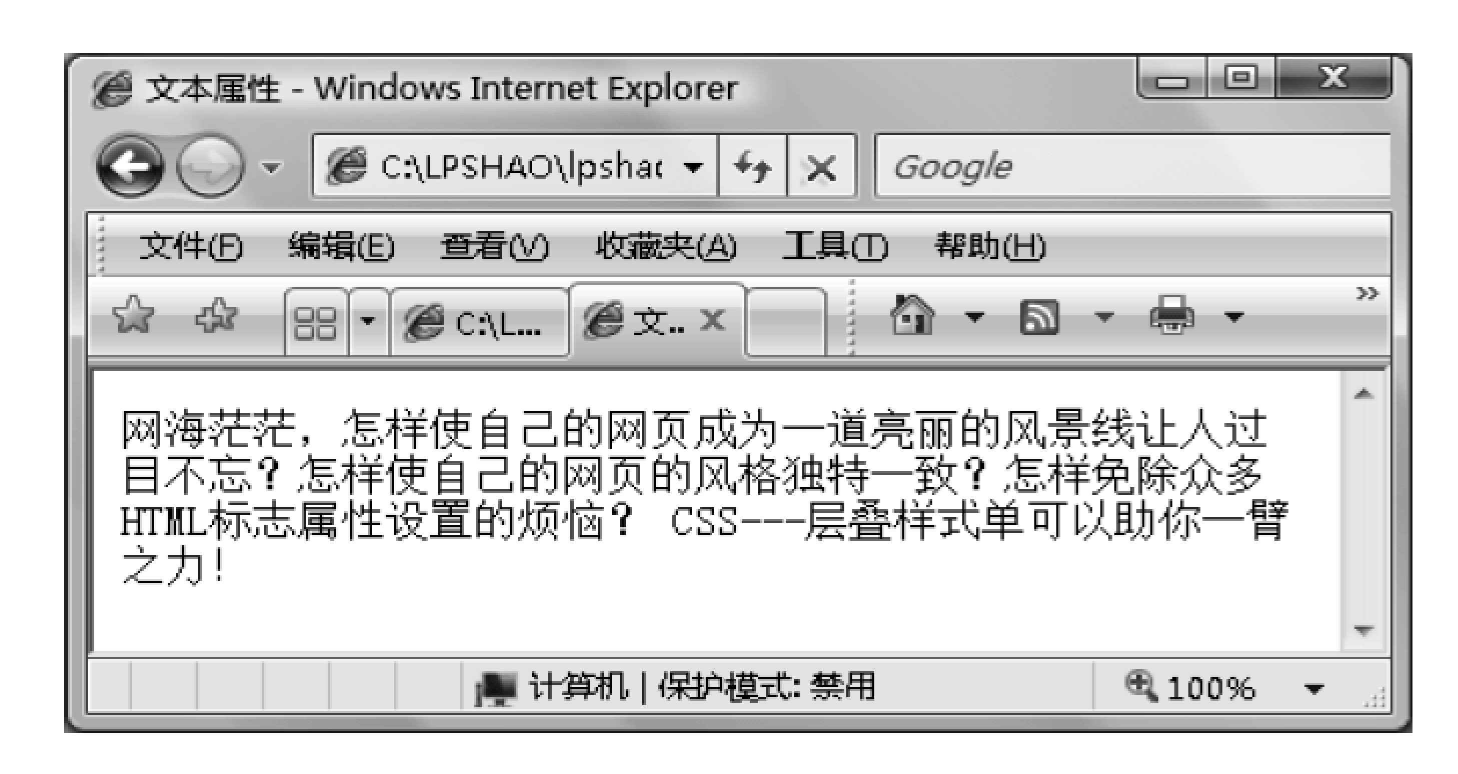


图 4.8　排版前网页 4-8.htm 的显示结果

例 4.9　修改 4-8.htm 文件代码，添加 CSS 文本属性代码对文本文字进行排版的文档（4-9.htm）。

```
<html>
<head>
<title>文本属性</title></head><body>
<p style="letter-spacing:1em;text-align:justify;text-indent:4em;line-
height:17pt">
网海茫茫，怎样使自己的网页成为一道亮丽的风景线让人过目不忘？
怎样使自己的网页的风格独特一致？
怎样免除众多 HTML 标志属性设置的烦恼？CSS——层叠样式单可以助你一臂之力！</p>
</body>
</html>
```

在浏览器打开 4-9.htm 网页，可以看到如图 4.9 所示的页面效果。

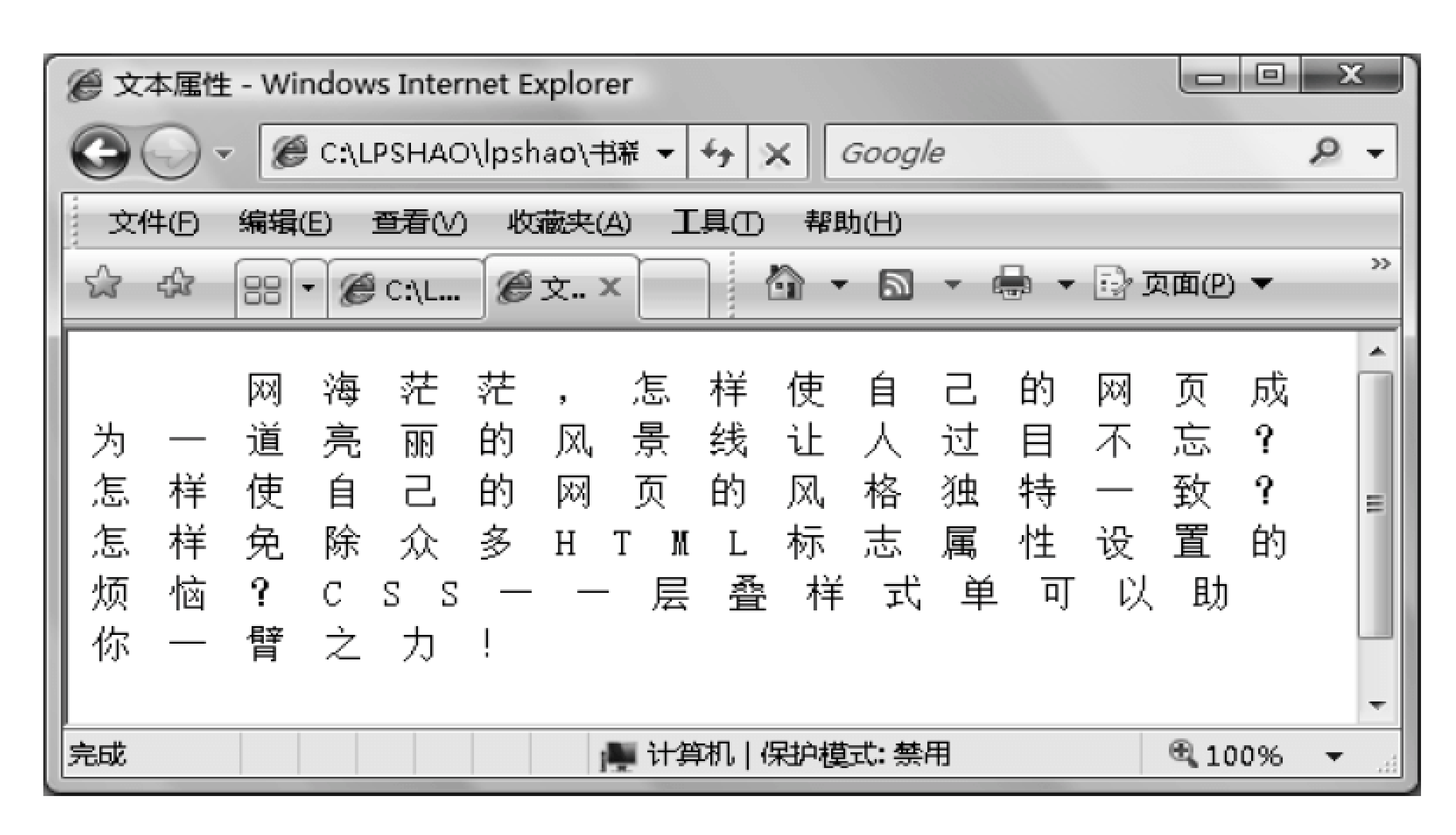


图 4.9　排版后网页 4-9.htm 的显示结果

2. 知识点归纳

（1）排版属性。从图 4.9 可以看到经过排版属性处理的文本，其中使用了字间距、行间距、文本对齐、首行文字缩进方式等属性。在 HTML 文件中使用排版属性的方式参考

如下代码：

```
<p style="letter-spacing:1em;text-align:justify; text-indent:4em; line-height:17pt">
…</p>
```

其中，字间距属性 letter-spacing 定义字间距为 1em；文本对齐属性 text-align 定义为两端对齐方式；首行文本缩进属性 text-indent 定义缩进 4em；行高属性 line-height 定义行高为 17pt。

（2）CSS 对文本进行排版的主要属性如表 4.3 所示。

表 4.3　对文本进行排版的主要属性

属　　性	属 性 含 义	属　性　值
Word-spacing	定义各个单词之间的间距	Normal<length>（必须以长度为单位）
Letter-spacing	定义每个字母之间的间距	Normal<length>（必须以长度为单位）
Text-decoration	定义文字的“装饰”样式	None | underline | overline | line-through | blink
Vertical-align	定义元素在垂直方向上的位置	Baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage>
Text-transform	使文本转换为其他形式	Capitalize | uppercase | lowercase | none
Text-align	定义文字的对方方式	Left | right | center | justify
Text-indent	定义文本的首行的缩进方式	<length> | <percentage>
Line-height	定义文本的行高	Normal | <number> | <length> | <percentage>

从表 4.3 中可以看到 CSS 可以定义文本的字间距、字母间距、装饰方式、对齐方式、缩进方式和行高等属性。

4.3.5　超链接属性

网页默认的超链接文字显示方式是：未访问过的超链接是蓝色文字带蓝色下划线；访问过的超链接是深紫色文字带深紫色下划线。

如果所有的网页都是这种样式会很单调，能否改变超链接文字的颜色呢？可以，利用 CSS 文本属性中的 Text-decoration 属性可以修改超链接文本的属性。

1. 重新定义超链接文本的属性

例 4.10　编写 HTML 文档：通过 CSS 的超链接文本属性 text-decoration 实现，没有访问过的超链接是绿色文字无下划线；访问过的超链接是红色文字无下划线。另外，当鼠标指定超链接时，超链接文字变为蓝色并添加上划线(4-10.htm)。

```
<html>
  <head>
  <style>
```

```
        a:link{color:green;text-decoration:none}
        a:visited{color:red;text-decoration:none}
        a:hover{color:blue;text-decoration:overline;font-size:26pt}
        p:{font-family:行书体;font-size:18pt}
    </style>
    </head>
    <body>
    <p><a href="http://www.263.net">未访问的链接</a></p>
    <p><a href="http://www.163.net">访问过的链接</a></p>
    <p><a href="http://www.sina.com">鼠标激活的链接</a></p>
    </body>
</html>
```

在浏览器中打开 4-10.htm 网页，可以看到如图 4.10 所示的页面效果。

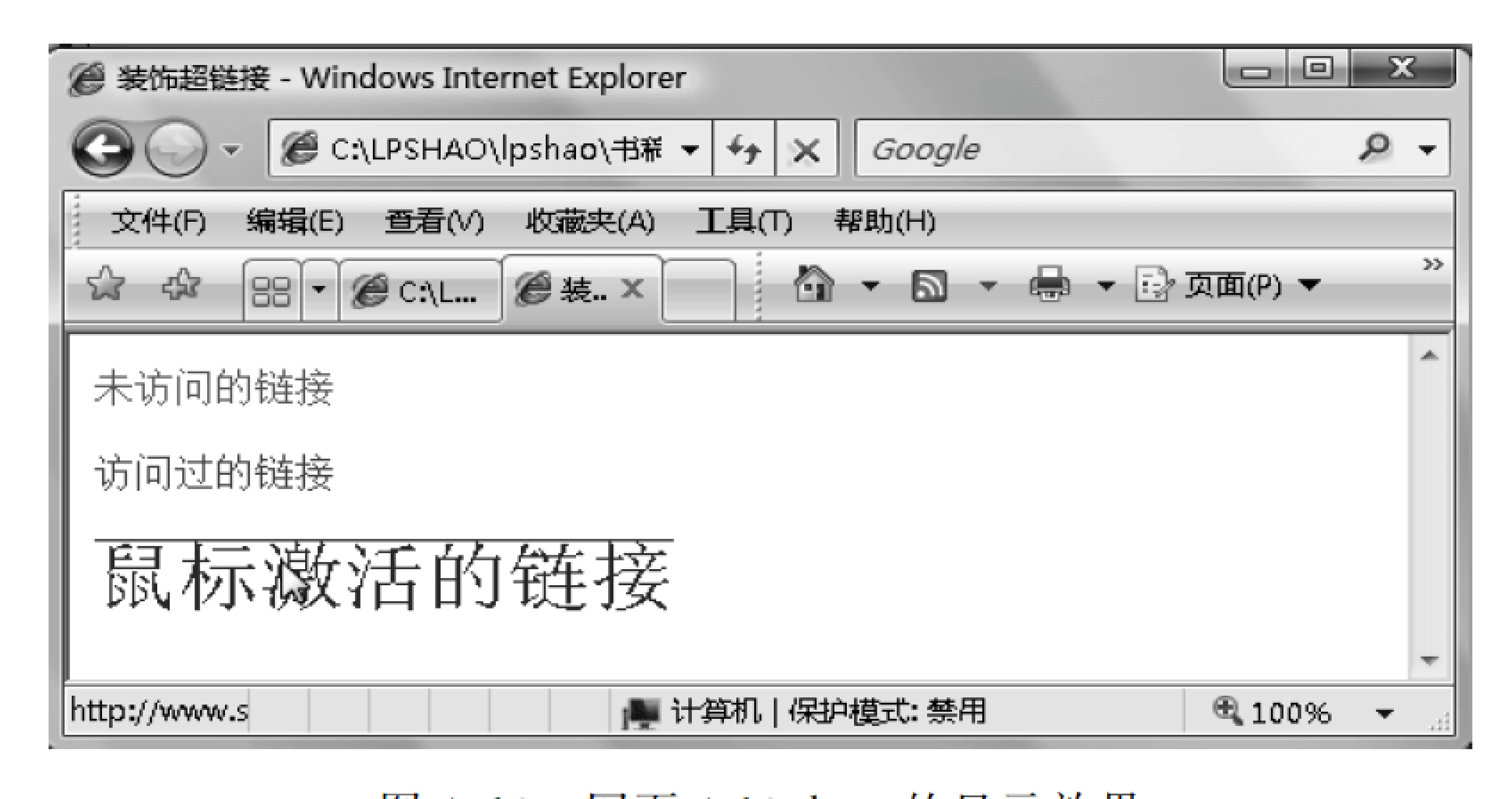


图 4.10　网页 4-10.htm 的显示效果

2. 知识点归纳

从本例中可以看出，CSS 样式可以嵌套使用，一起对对象元素起作用。

4.3.6　边距属性

CSS 可以定义 margin 边距属性。通过给标记码指定 margin 属性可定义标识的对象与其他对象的边距。

1. 定义边距属性

例 4.11　定义标记码<p>的 margin 边距属性，设置其上、右、下、左的边距分别为 1em、2em、3em、4em 的文档(4-11.htm)。

```
<html>
<head>
<style>
  p{margin:1em 2em 3em 4em}
```

```
</style>
</head>
<body>
网海茫茫,怎样使自己的网页成为一道亮丽的风景线让人过目不忘?
怎样使自己的网页的风格独树一帜?
<p>
  怎样免除众多 HTML 标志属性设置的烦恼?
</p>
  CSS---层叠样式单可以助你一臂之力!
</body>
</html>
```

在浏览器中打开 4-11.htm 网页,可以看到如图 4.11 所示的页面效果。可以看到具有 margin 属性的文本对象与其他文本、窗口边框的边距分别为 1em、2em、3em 和 4em。

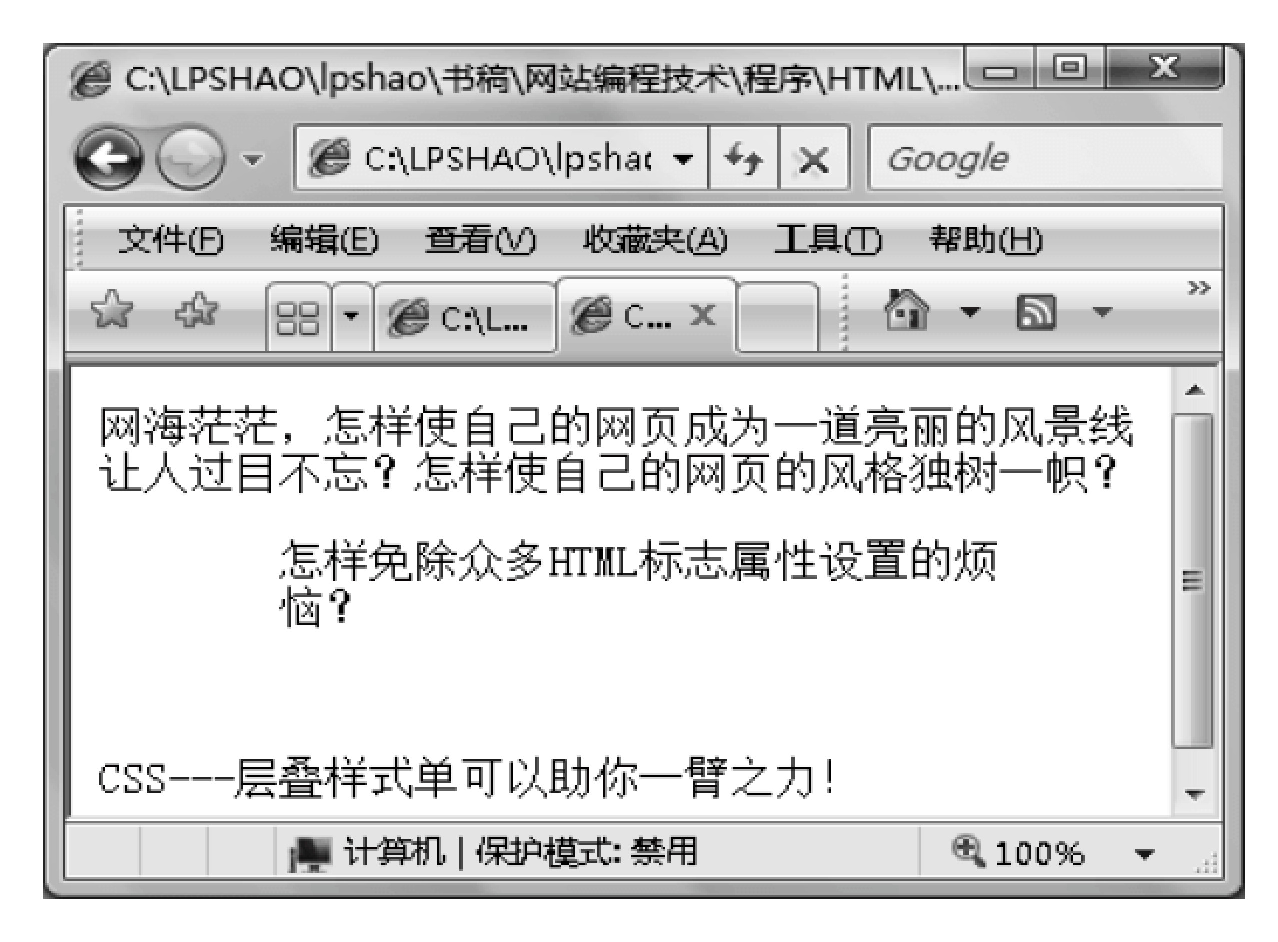


图 4.11　网页 4-11.htm 的显示效果

2. 知识点归纳

(1) margin 具有四种边距属性,如表 4.4 所示。

表 4.4　边距属性

属　　性	属 性 含 义	属　性　值
Margin-top	设置顶端边距	Length|percentage|auto
Margin-right	设置右侧边距	Length|percentage|auto
Margin-bottom	设置底端边距	Length|percentage|auto
Margin-left	设置左侧边距	Length|percentage|auto

(2) 具有 margin 属性的对象,无论怎么改变窗口的大小,都会按照定义好的边距样式在页面中显示。

(3) 简化定义 margin 属性时可以使用完整方式，即“属性：属性值”，也可以采用简化定义方式，即“margin：属性值列”，属性值按照上、右、下、左顺序依次排列。例如：

```
body{margin:1em 2em 3em 4em}
```

上面这段代码的含义是 body 的上、右、下、左边距为 1em、2em、3em、4em。

如果“属性值列”里有缺失的属性值，仍然按照上、右、下、左边距顺序对应选取。例如 body{margin：1em 2em}为上、右边距为 1em、2em。

4.3.7 边框的填充距属性

1. 填充距与边距的区别

边框为线条构成的矩形框。填充距指的是边框与框中对象之间的距离。

边距指的是具有 margin 属性的对象与页面中其他对象(例如文本、图像、窗口边框)之间的距离。

2. 设置边框与填充距属性

例 4.12 通过 CSS 的 padding 属性设置边框与框中文本对象之间的左、右、上、下填充距分别为 1em、2em、3em、6em 的文档(4-12.htm)。

```
<html>
<head>
<style>
p{border:2px ridge #FF0000;
padding-left:6em; padding-right:2em; padding-top:1em; padding-bottom:3em}
</style>
</head>
<body>
网海茫茫，怎样使自己的网页成为一道亮丽的风景线让人过目不忘？
怎样使自己的网页的风格独树一帜？
<p>怎样免除众多 HTML 标志属性设置的烦恼？
CSS---层叠样式单可以助你一臂之力！
</p>
</body>
</html>
```

在浏览器中打开 4-12.htm 网页，可以看到如图 4.12 所示的页面效果。

3. 知识点归纳

1) padding 填充距属性

padding 填充距属性如表 4.5 所示。

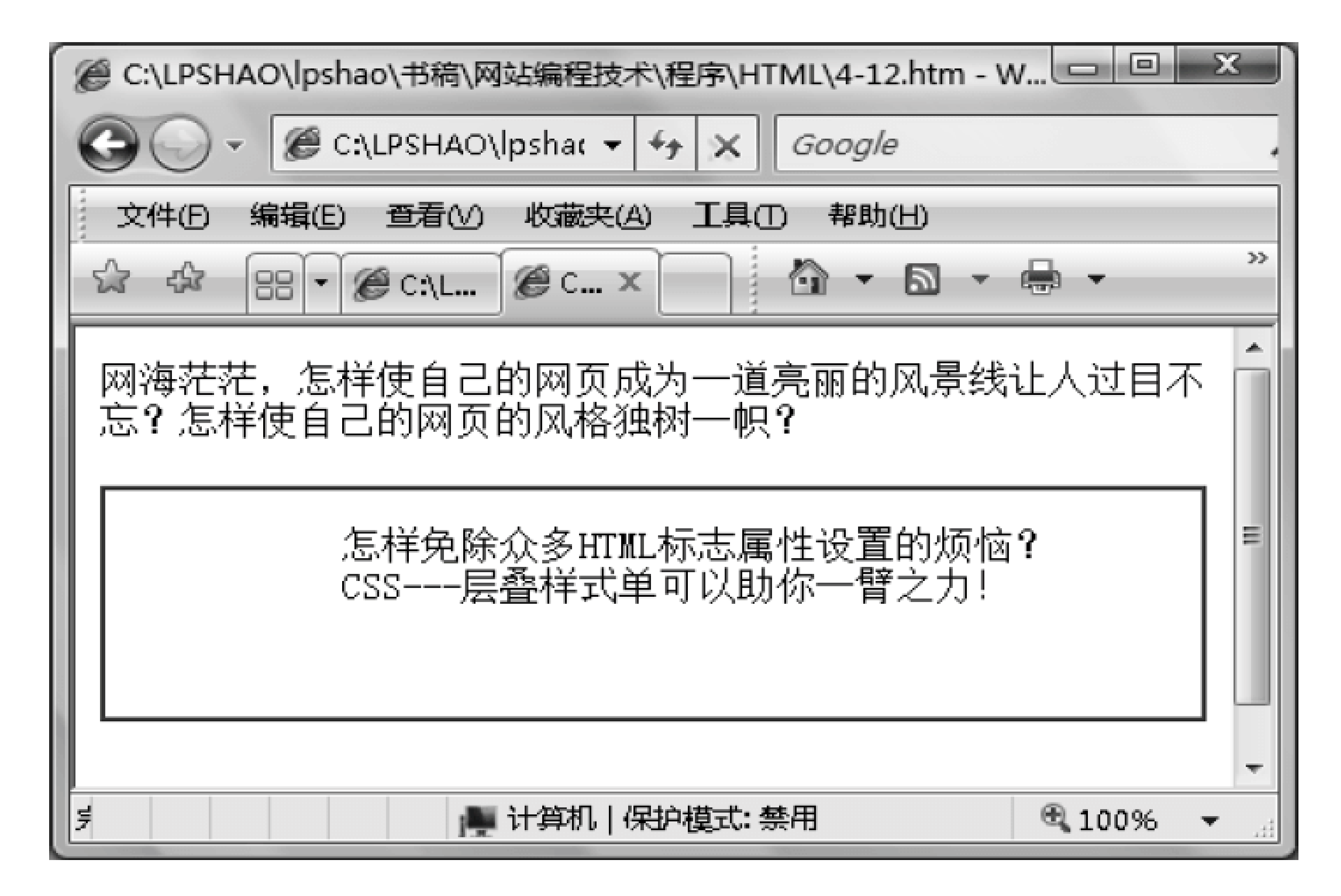


图 4.12 网页 4-12.htm 的显示效果

表 4.5 填充距属性

属　　性	属 性 含 义	属　性　值
Padding-top	设置顶端填充距	Length|percentage
Padding-right	设置右侧填充距	Length|percentage
Padding-bottom	设置底端填充距	Length|percentage
Padding-left	设置左侧填充距	Length|percentage

2）border 边框属性

通过 CSS 的 border 属性可以设置边框的宽度、样式和颜色。border 边框属性如表 4.6 所示。

表 4.6 边框属性

属　　性	属 性 含 义	属　性　值
Border-top-width	设置顶端边框宽度	Thin|medium|thick|length
Border-right-width	设置右侧边框宽度	Thin|medium|thick|length
Border-bottom-width	设置底端边框宽度	Thin|medium|thick|length
Border-left-width	设置左侧边框宽度	Thin|medium|thick|length
Border-width	一次定义边框宽度	Thin|medium|thick|length
Border-color	设置边框颜色	Color
Border-style	设置边框样式	None | dotted | dash | solid | double | groove|ridge|inset|outset
Border-top	一次定义顶端的各种属性	Border-top-width|border-style|color
Border-right	一次定义右侧的各种属性	Border-top-width|border-style|color
Border-bottom	一次定义底端的各种属性	Border-top-width|border-style|color
Border-left	一次定义左侧的各种属性	Border-top-width|border-style|color

边框属性种类繁多，读者可以通过练习来了解不同属性的样式类型。

4.3.8 图文混排属性

在页面中插入图片很难控制图片对象与文本对象之间的距离，使用表格分别安排文字与图片对象是一种解决方案，但表格会影响页面的下载速度。使用 CSS 的边距属性、float 等属性可以轻松解决页面中图文之间的距离问题。

1. 图文混排的效果

例 4.13 通过 CSS 的多个不同属性实现图文混排效果的文档(4-13.htm)。

```
<html>
<head>
<style>
img {margin-right:1em; margin-bottom:1em; float:left; width:100; height:80}
body{text-align:left; text-indent:2em; line-height:20pt}
</style>
</head>
<body>
<p><img border="0" src="鸡.gif" width="120" height="90">网海茫茫，怎样使自己的网页成
为一道亮丽的风景线让人过目不忘?
怎样使自己的网页的风格独树一帜?
怎样免除众多 HTML 标志属性设置的烦恼?
CSS---层叠样式单可以助你一臂之力!</p>
</body>
</html>
```

在浏览器中打开 4-13.htm 网页，可以看到如图 4.13 所示的页面效果。

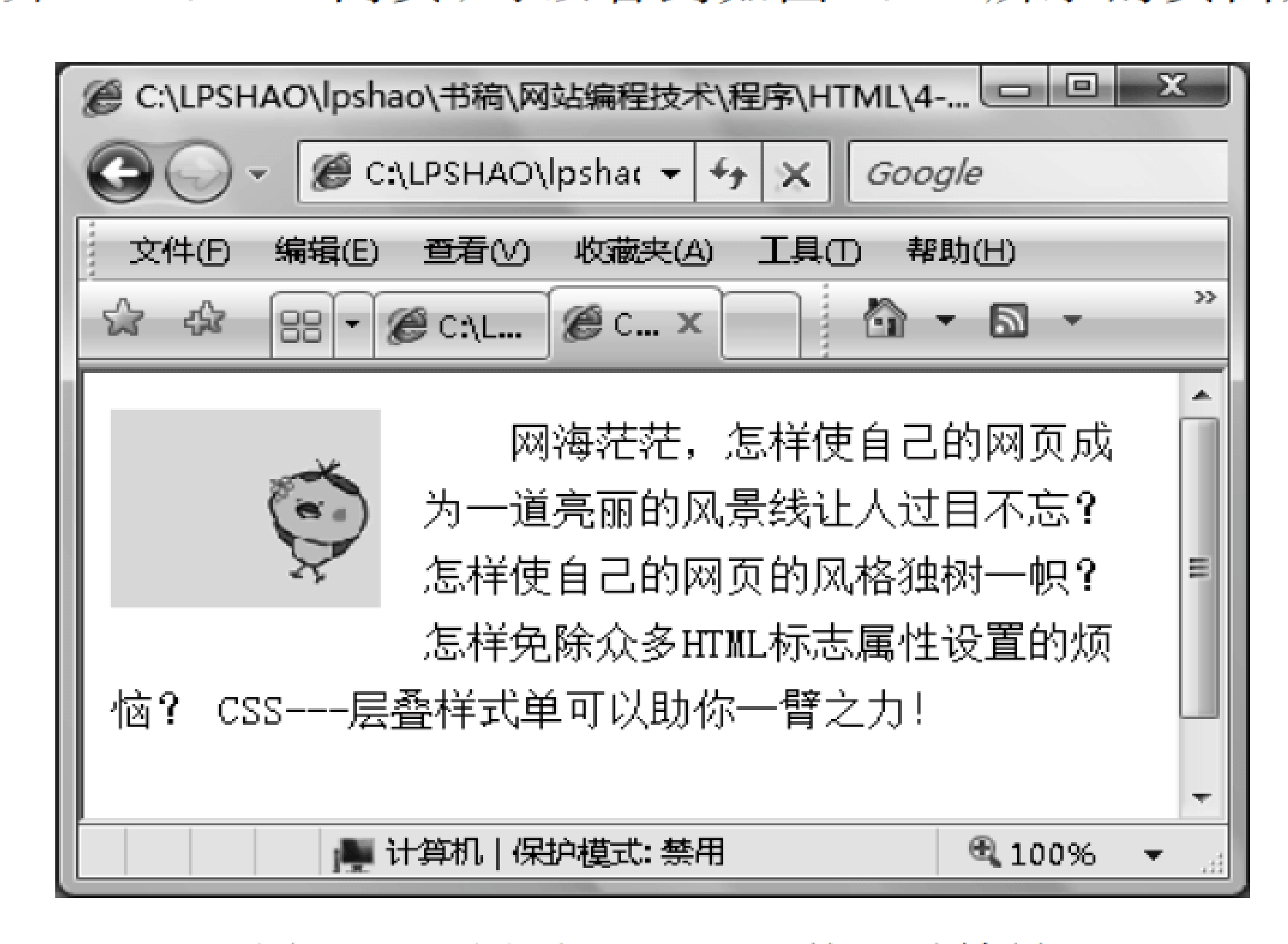


图 4.13 网页 4-13.htm 的显示效果

2. 知识点归纳

1）高度、宽度、float 属性

4-13.htm 中使用了 CSS 高度、宽度、float 等属性，如表 4.7 所示。

表 4.7　CSS 的其他属性

属　性	属 性 含 义	属　性　值
Width	定义宽度属性	Length|percentage|auto
Height	定义高度属性	Length|auto
Float	使文字环绕在一个元素的四周	Left|right|none
Clear	定义某一边是否有环绕文字	Left|right|none|both

其中，通过宽、高属性可以定义指定元素的大小，如果不指定元素的大小其可以随着窗口的改变自动改变其大小。float 属性可以实现文字环绕图片的混排效果。

2）使用 CSS 样式设置图文混排效果的优势

从 4-13.htm 中可以看出不用表格也能实现图文混排的效果，与使用表格相比减少了大量的代码，更可加快下载速度。使用 CSS 样式设置图文混排效果会使图文布局更合理，页面更漂亮。

4.3.9　位置属性与可视性属性

如何确定显示对象在页面中的具体位置呢？可使用 CSS 的 position 位置属性解决这个问题。

1. 相对位置、绝对位置与静态位置

在使用 CSS 位置属性之前，先要了解 position 位置属性的三个基本概念，相对位置、绝对位置与固定位置。

相对位置是指对象元素的位置是相对于未定义情况下它应在位置的偏移位置，偏移量为相对定义中的数值。

绝对位置是指对象元素与其所在容器块中保持的位置。通常情况下元素的容器块为页面边界，即相对页面边界的固体位置。

静态位置是指对象元素在窗体页面上的位置是不可修改的，对于那些没有被指定位置属性的元素，它是默认值。

2. 可视性

通过 visibility 属性可以定义在页面上显示的元素对象是否可以可见，visibility 属性只有两个值 hidden（隐藏）与 visible（可见）。

3. 使用位置属性与可视性属性

例 4.14 通过 CSS 的位置属性确定图片在页面中的绝对位置，并可根据用户的选择在同一位置显示不同图片的文档(4-14.htm)。

```
<html>
<head>
<style type="text/css">
        #c1{position:absolute; top:100; left:30}
        #c2{position:absolute; top:100; left:30; visibility:hidden;}
        p{color:#0000FF; font-weight:bold}
</style>
</head>
单击按钮会显示不同的图片
<form name="myform">
<input type="button" value="鲜花" onclick="c1.style.visibility='visible'; c2.style.
visibility='hidden'">
<input type="button" value="小鸡" onclick="c2.style.visibility='visible'; c1.style.
visibility='hidden'">
</form>
<div id="c1"><img src="花.gif" width="120" height="90"><p>鲜花图像</div>
<div id="c2"><img src="鸡.gif" width="120" height="90"><p>小鸡图像</div>
</html>
```

在浏览器中打开 4-14.htm 网页，可以看到如图 4.14 所示的页面效果。单击不同的按钮可以在同一位置显示不同的图片。



图 4.14　网页 4-14.htm 的页面效果

4. 知识点归纳

1）常用的定位属性

位置属性是用来定位的，用来定位的属性还有其他属性如表 4.8 所示。

表 4.8　定位属性

属　性	属 性 含 义	属　性　值
Position	定义位置	Absolute|relative|static
Left、top	指定横向、纵向坐标位置	Length|percentage|auto
Width、height	指定占用空间的大小	Length|percentage|auto
Clip	剪切	Shape|auto
Overflow	内容超出时的处理方法	Visible|hidden|scroll|auto
z-index	产生立体效果	Auto|integer
Visibility	定义可见性	Inherit|visible|hidden

2）可视行属性及 ID 类型样式的特殊性

从 4-14. htm 可以看到通过 ID 名可以定义其样式的属性，例如下面的语句：

```
c1.style.visibility='hidden'; (隐藏)
onclick="c2.style.visibility='visible'" (可视)
```

通过 ID 名可以直接设置其样式的属性，修改属性的值。这就是使用 ID 名来定义 CSS 样式的一个用途。

4.3.10　空间位置属性

通过 CSS 位置属性可以确定元素在页面中的位置。但在相同的位置怎样显示多个对象呢？如何确定它们的层叠次序呢？通过 z-index 属性可以解决这个问题。

1. 使用 z-index 属性

例 4.15　利用 z-index 属性实现两幅图片和一段文字在相同页面位置、不同空间位置（不同层次）显示，文字对象覆盖在 W2 图片对象上，“铃”图片对象覆盖在文字对象上的文档（4-15. htm）。

```
<html>
<head>
<title>空间位置效果</title>
<style type="text/css">
    .p1{position:absolute; left:1in; top:0in; width:3in; height:2in}
    .p2{position:absolute; left:2.4in; top:0.2in}
```

```
</style>
</head>
<body>
  <img src="img/w2.jpg" class="p1" id="image" style="z-index:1">
  <div class="p1" id="text1" style="color:#ffff33; z-index:2" >
    这段文字会覆盖在图片上。因为文字和 w2 图片属于同一类 p1,因此具有相同的位置设置。
w2 图片 zindex 值为 1 位于最底层;铃图片的 z-index 值为 3 位于最上层
  </div>
  <img src="img/铃.gif"  class="p2" id="image" style="z-index:3">
</body>
</html>
```

在浏览器中打开 4-15.htm 网页,可以看到如图 4.15 所示的页面效果。

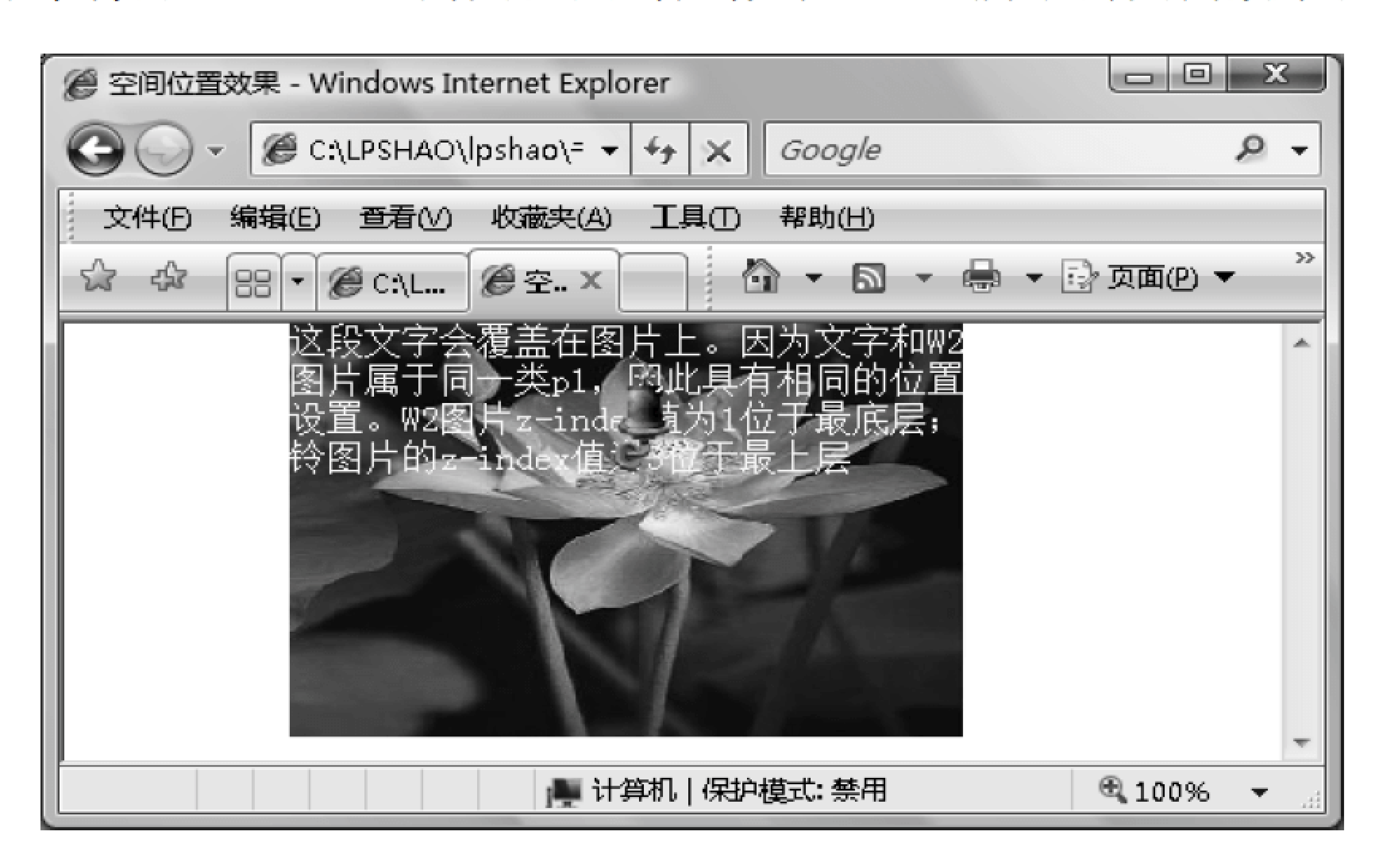


图 4.15　网页 4-15.htm 的页面效果

2. 知识点归纳

1) z-index 属性

z-index 属性通过其属性值确定对象在页面上叠放的层次位置,值越小位于的层次越低。

利用 CSS 的定位属性,可以确定对象在页面上的平面位置与空间位置,会使页面更加精致,更加富有动感。

2) 通过网页编辑工具设置 CSS 样式

学习 CSS 的属性可以借助网页编辑工具,大多数的网页编辑工具都支持 CSS,可以自动编写 CSS 代码。例如,打开 FrontPage 2003,打开网页文件,在网页"设计"视图下,单击主窗口"格式"→"样式"菜单项,可打开"样式"对话框,如图 4.16 所示,在对话框中可以选中已有的样式进行修改,也可以新建样式,进行设置后 FrontPage 2003 会自动编写相应的 CSS 代码。

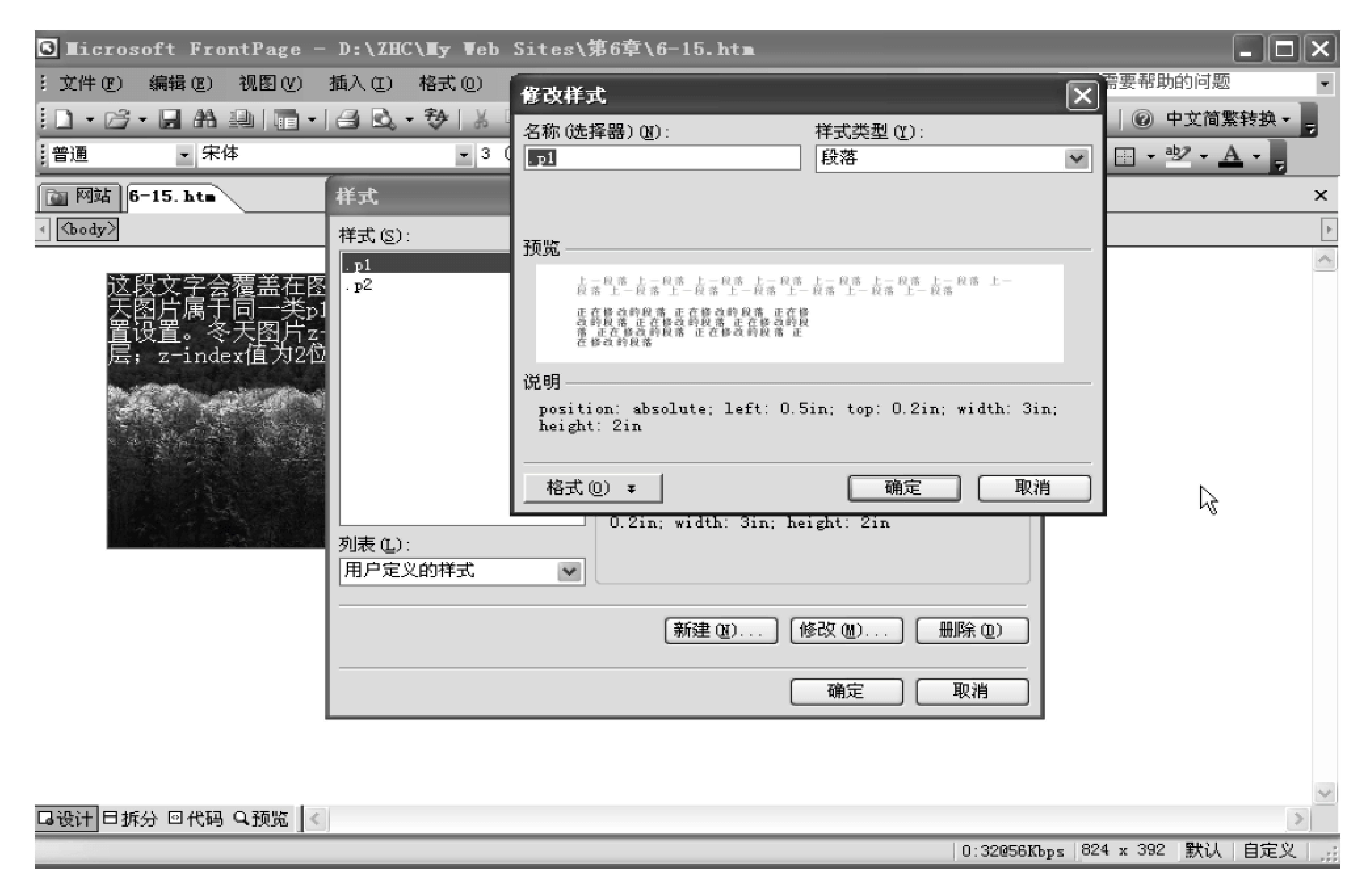


图 4.16　使用工具创建与修改 CSS 样式

4.4　使用 CSS 滤镜属性使对象显示特殊效果

CSS 最精彩的内容是滤镜，通过滤镜属性 filter 可以对对象进行滤镜处理，使对象产生特殊的显示效果。

本节的内容主要解决如何通过 CSS 滤镜属性 filter 的子属性对对象进行滤镜效果处理的问题。

4.4.1　透明效果

透明度不同，图像会显示不同的效果，如何使图像显示不同的透明度呢？可使用滤镜属性 filter 的 alpha 子属性解决这个问题。

1. 使用 alpha 透明度属性

例 4.16　通过 alpha 属性设置图像不同透明效果的文档(4-16.htm)。

```
<html>
  <head>
    <title>alpha 透明属性</title>
    <style>
    #img1{filter:alpha(opacity=40)}
    #img2{filter:alpha(opacity=0,finishopacity=100,style=1,startX=0,startY=85,
finishX=150,finishY=85)}
```

```
    # img3 {filter: alpha (opacity= 0, finishopacity= 100, style= 2, startX= 0, startY= 85,
finishX= 150, finishY= 85) }
    # img4 {filter: alpha (opacity= 0, finishopacity= 100, style= 3, startX= 0, startY= 85,
finishX= 150, finishY= 85) }
    </style>
  </head>
  <body>
    <img src= "img/w2.jpg" width= "120" height= "90" >
    <img ID= "img1" src= "img/w2.jpg" width= "120" height= "90">
    <img ID= "img2" src= "img/w2.jpg" width= "120" height= "90">
    <img ID= "img3" src= "img/w2.jpg" width= "120" height= "90">
    <img ID= "img4" src= "img/w2.jpg" width= "120" height= "90">
  </body>
</html>
```

在浏览器打开 4-16.htm 网页，可以看到如图 4.17 所示的页面效果。其中，左一为原始图像，左二为透明度为 40 的图像，左三透明度为线形渐变的图像，左四透明度为放射形渐变的图像，左五透明度为长方形渐变的图像。



图 4.17　网页 4-16.htm 的页面效果

2. 知识点归纳

1）alpha 透明子属性的语法格式

要了解 alpha 是如何设置透明度的，先来看一下使用滤镜属性 filter 的 alpha 子属性设置透明度的语法格式：

```
filter: alpha (opacity= n, finishopacity= m, style= s, startX= x0, startY= y0, finishX= x1,
finishY= x2)
```

2）alpha 透明子属性的参数

- opacity 参数代表透明度等级，参数值从 0 到 100，0 代表完全透明，100 代表完全不透明。
- finishopacity 参数是可选项，用来设置结束时的透明度，从而达到一种渐变效果，参数值从 0 到 100。

- style 参数指定透明区域的形状特征。0 统一形状，1 线形，2 放射状，3 长方形。
- startX 和 startY 代表渐变透明效果的开始坐标，finishX 和 finishY 代表渐变透明效果的结束坐标。

如果不设置透明度的渐变效果，只需设置 opacity 参数即可。

4.4.2 模糊效果与投影效果

滤镜属性 filter 的 blur 子属性可以使文本对象产生模糊效果，滤镜的 dropshadow 属性可以使文本对象产生投影效果。看下面的例子。

1. 使用 blur 与 dropshadow 子属性

例 4.17 通过设置 blur 与 dropshadow 属性使文本对象产生模糊与投影效果的文档(4-17.htm)。

```
<html>
<head>
<style>
#f1{position:absolute; top:10;width:300; filter:blur(add=true,direction=136,strength=
30)}
#f2{position:absolute; top:10; left:3in; width:300; filter:dropshadow(color=#ffccff,
offx=16,offy=10,positive=0)}
</style>
</head>
<div id=f1 style="font-family:宋体; font-size:66; font-weight:bold; color:#cc00cc;">
你好</div>
<div id=f2 style="font-family:宋体; font-size:66; font-weight:bold; color:#cc00cc;">
你好</div>
</html>
```

在浏览器中打开 4-17.htm，可以看到如图 4.18 所示的页面效果。

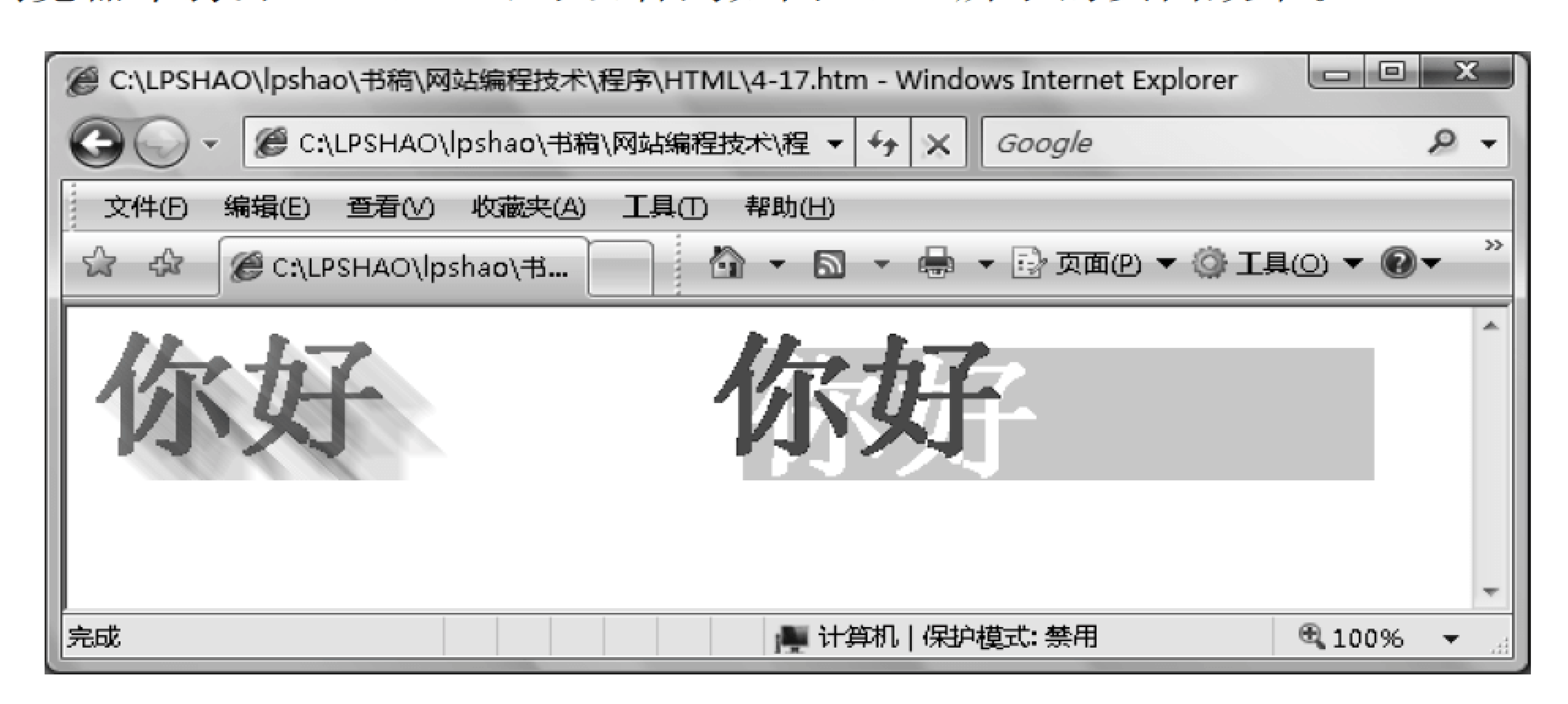


图 4.18 网页 4-17.htm 的页面效果

2. 知识点归纳

1）dropshadow 属性的语法格式

```
filter:blur(add=1,direction=d,strength=int)
```

blur 子属性能使元素产生一种模糊效果，blur 具有 3 个参数。add 参数有两个值：True 和 False。分别指定元素（图像、文字）是否被设置成模糊效果；direction 参数用来设置模糊的方向，0 度代表垂直向上，以 45 度为一个单位，例子中的 135 代表底部向右 135 度，每一个度数单位都代表一个模糊方向；strength 代表有多少像素的宽度将受到模糊影响，参数值是用整数来设置的。

2）dropshadow 属性的语法格式

```
filter:dropshadow(color=color, offx=offx, offy=offy, positive=positive)
```

Dropshadow 子属性能使元素产生一种投影效果，Dropshadow 具有 4 个参数，color 代表投射投影的颜色。offx 和 offy 分别为 x 方向和 y 方向投影的偏移量，偏移量必须用整数值来设置，如果设置为正整数，代表 x 轴的右方向和 y 轴的向下方向。设置为负整数则相反。positive 参数有两个值，true 或 1 用非透明像素建立可见的投影，false 或 0 用透明的像素建立可见的投影。

3）注意问题

dropshadow 属性对图片的支持不好，因为 JPEG、GIF 格式的图像文件经过了减色和压缩处理，颜色很丰富，很难找到一个投射投影的位置。

4.4.3 发光效果

滤镜属性 filter 的 glow 子属性可以使对象的边缘产生发光效果，怎样对文本对象或图片对象应用 glow 属性呢？

1. 使用 glow 发光属性

例 4.18 使用 glow 属性使文本对象与图片对象产生发光效果的文档（4-18.htm）。

```
<html>
<head>
<style>
#f{position:absolute; left:6; top:9; width:300; filter:glow(color=red,strength=26)}
//* 定义 f 样式，采用绝对位置，Glow 子属性，发光颜色值为#FF3399，强度为 26* //
</style>
</head>
<body>
<div id=f style="width:688; height:270">
<p style="font-family:宋体; font-size:66pt; font-weight:bold; color:#003366">
你好！
<img border="0" src="img/铃.gif" width="32" height="48"></p>
```

```
</div>
</body>
</html>
```

在浏览器中打开 4-18.htm 网页,可以看到如图 4.19 所示的页面效果。

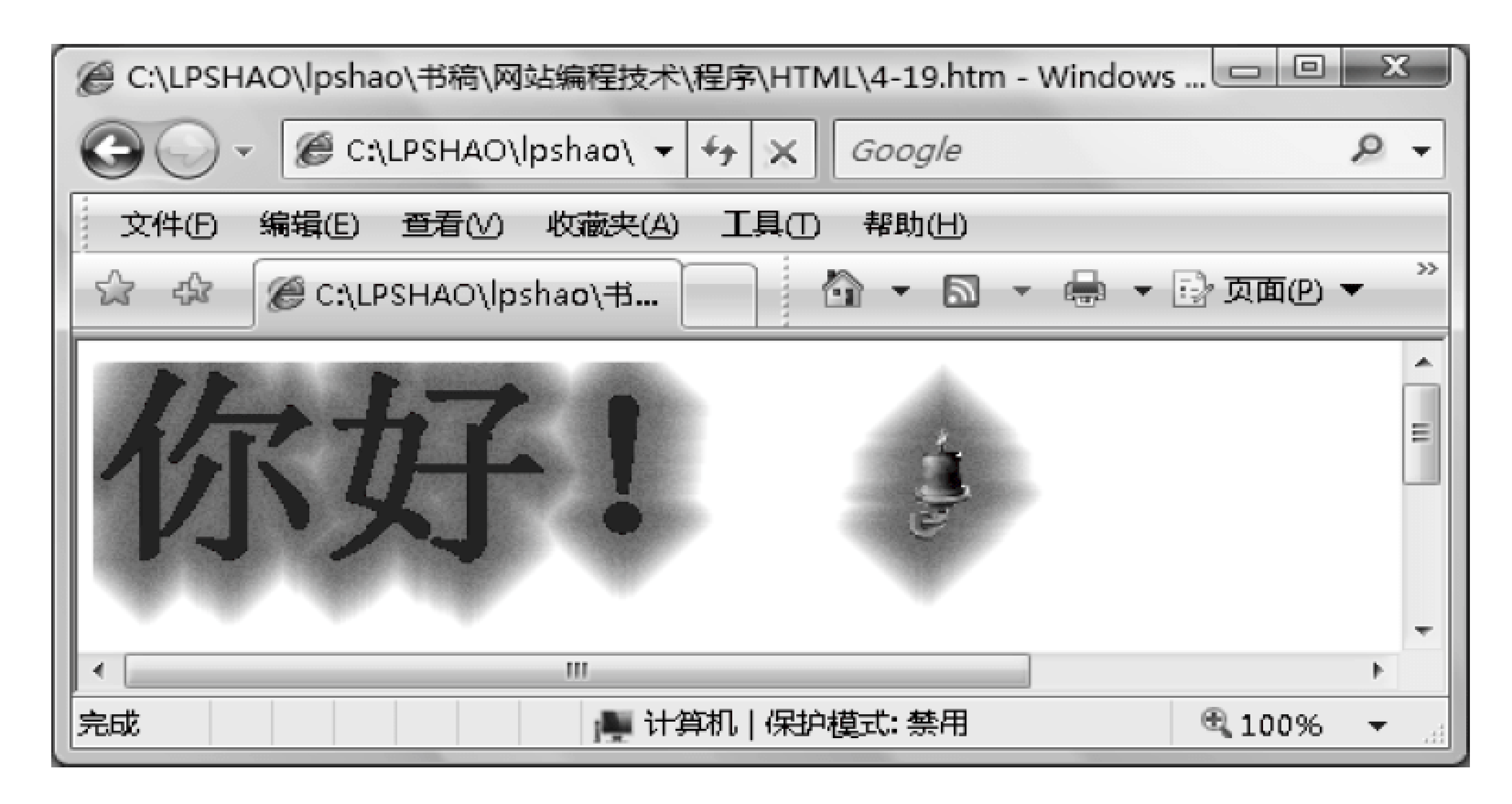


图 4.19　网页 4-18.htm 的页面效果

2. 知识点归纳

1) glow 发光属性的语法格式

```
Filter:glow(color=color, strength=strength)
```

glow 属性参数 color 用来确定发光的颜色,strength 用来确定发光的强度,参数值从 1 到 266。

2) 注意问题

glow 发光属性对图像对象、文本对象都可以使用。通过修改 color 参数值,可以显示不同颜色的发光效果。

4.4.4　波纹效果

滤镜属性 filter 的 wave 子属性可以将对象元素的显示样式按照垂直的波纹样式打乱,wave 属性能使文本对象或图片对象产生什么样的波纹效果呢?

1. 应用 wave 波纹属性

例 4.19　设置 wave 属性使文本对象与图片对象产生波纹效果的文档(4-19.htm)。

```
<head>
<style>
#w1{position:absolute; top:10; width:300; filter:wave(add=true,freq=3,lightstrength=
100,phase=66,strength=20)}
//* 设置 w1 样式,绝对位置,wave 属性产生 3 个波纹,光强为 100,波纹从 162 度(360 * 66%)开始,
振幅为 20 * //
```

```
</style>
</head>
<body>
<div id=w1 style="width:369; height:173; z-index:2;">
<p style="font-family:lucida handwriting; font-size=72pt; font-weight:bold; color:rgb
(189,1,66);">
Winter</p>
</div>
<img id=w1 src="img/w1.jpg" style="width:369; height:173; z-index:1;" >
</body>
</html>
```

在浏览器打开 4-19.htm 网页，可以看到如图 4.20 所示的页面效果。

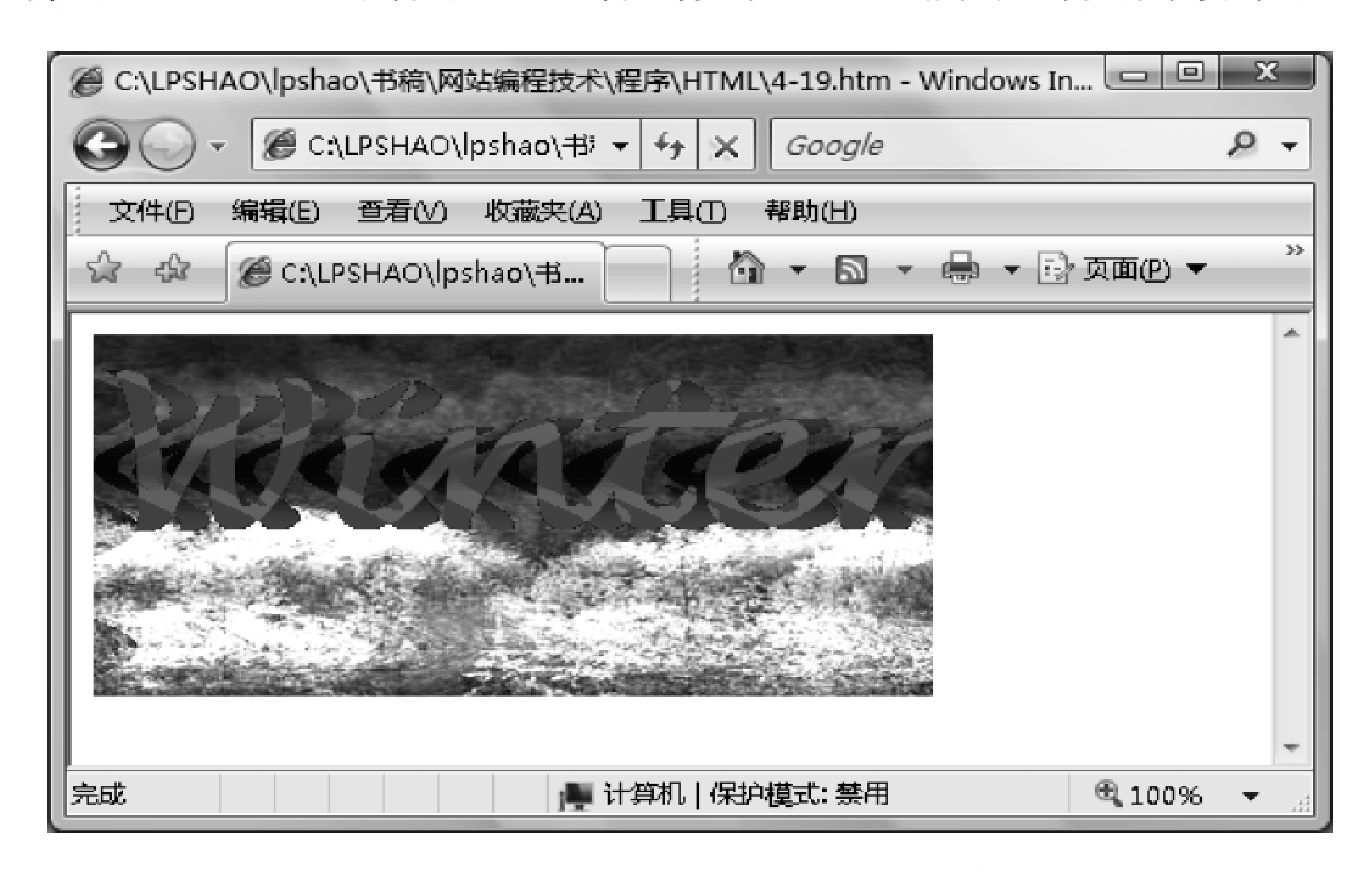


图 4.20　网页 4-19.htm 的页面效果

2. 知识点归纳

1）wave 属性的语法格式

filter:wave(add=true(false),freq=频率,lightstrength=增强光效,phase=偏移量,strength=强度)

2）wave 属性的 5 个参数

- add 参数用来定义是否产生波纹，其有两个参数值，true 表示生产波纹样式，false 不产生。
- freq 参数用来定义生成波纹的频率，是指在对象上共产生多少个完整的波纹。
- lightstrength 参数用来使生成的波纹增强光的效果，参数值可以为 0～100。
- phase 参数用来设置正弦波开始的偏移量，默认值为 0，范围为 0～100，值表示波纹开始时偏移量占波长的百分比，例如该值为 26，代表正弦波从 90 度（360 * 26%）的方向开始偏移。

- Strength 指振幅强度。

4.4.5 滤镜的其他属性

1. chroma 透明色属性

滤镜属性 filter 的 chroma 子属性可以将一个对象的颜色设置为透明色，它的语法格式如下：

```
filter:chroma(color=color)
```

chroma 透明色属性只有一个参数 color。例如，用下面的代码可以把 div 范围内的绿色设置为透明色。

```
div{position:absolute; top:70; width:200; filter:chroma(color=green)}
```

2. flipH、flipV 水平与垂直翻转属性

滤镜属性 filter 的 flip 子属性可以设置对象的翻转样式，flipH 代表水平翻转，flipV 代表垂直翻转。它们的语法格式为：

```
filter:flipH
filter:flipV
```

例如，用下面的代码可以把 div 范围内的对象水平翻转。

```
div{position:absolute; top:20; width:300; filter:flipH}
```

3. invert 翻转属性

滤镜属性 filter 的 invert 子属性可以把对象的可视化属性全部翻转，包括色彩、饱和度和亮度值。它的语法格式如下。

```
filter:invert
```

Invert 翻转属性实际上是一种“底片”的效果，读者可以尝试一下。

4. mask 遮照属性

滤镜属性 filter 的 mask 遮照子属性可以为对象建立一个覆盖于表面的膜。它的语法格式如下。

```
Filter:mask(color=颜色)
```

mask 遮照属性只有一个 color 参数，用来指定使用什么颜色作为掩膜。对象加上 mask 遮照属性后的效果就好像用有色眼镜看到的对象一样。例如，用下面的代码可以指定在 div 范围内用什么颜色遮住对象。

```
div{position:absolute; top:20; left:60; filter:mask(color:#666699)}
```

使用 mask 遮照属性，遮罩下的字体颜色就失去了意义。

5. shadow 投影属性

滤镜属性 filter 的 shadow 投影属性可以在指定的方向建立对象的投影。它的语法格式如下。

```
filter:shadow(color=color,direction=direction)
```

shadow 有两个参数值，color 参数用来指定投影的颜色。direction 参数用来指定投影的方向。这里说的方向与 blur 属性中提到的“方向与角度的关系”是一样的。

Dropshadow 子属性与 shadow 子属性有什么不同呢？shadow 属性可以在任意角度进行投影，dropshadow 属性实际上是用偏移来定义投影的。

6. xray 射线属性

滤镜属性 filter 的 Xray 子属性可以产生 X 射线。Xray 属性产生的效果使对象看上去有一种 X 光片的感觉。其语法格式如下。

```
filter:Xray
```

如果给对象设置了 Xray 射线属性就像给它拍了一张 X 光片一样，读者可以尝试一下。

从本节介绍的内容可以发现，filter 滤镜属性的语法格式有些特殊，它具有多个子属性，子属性又具有多个参数，filter 滤镜属性的语法格式可以归纳为：

```
filter: 子属性名 (子属性参数:子属性值;子属性参数:子属性值;…;子属性参数:子属性值;)
```

要进行滤镜处理必须使用 filter 属性，由子属性确定具体的滤镜处理方式，子属性名后括号内的子属性参数值确定滤镜的显示效果。Filter 的常用子属性如表 4.9 所示。

表 4.9 filter 的常用子属性及含义

属性名称	属 性 解 释	属性名称	属 性 解 释
Alpha	设置透明度	Grayscale	设置灰度(降低图片的彩色度)
Blur	设置模糊效果	Invert	设置底片效果
Chroma	设置指定颜色透明	Light	设置灯光投影
Dropshadow	设置投射阴影	Mask	设置透明膜
Fliph	水平翻转	Shadow	设置阴影效果
Flipv	垂直翻转	Wave	利用正弦波纹打乱图片
Glow	为对象的外边界增加光效	Xray	只显示轮廓

4.5 CSS 应用实例

本节的内容主要解决 CSS 综合应用的问题。

4.5.1 解释菜单

1. 创建 JS 文件

例 4.20 编写定义根据不同菜单显示不同文字使用的变量、函数和判别方式的 JavaScript 脚本文件,保存文件为“解释菜单.js”。

解释菜单.js 文件的代码如下:

```
msg=new Array(5)
msg[0]="<b>这是本章第一个文件 4-1.htm!</b>"
msg[1]="<b>这是本章第二个文件 4-2.htm!</b>"
msg[2]="<b>这是本章第三个文件 4-3.htm!</b>"
msg[3]="<b>首都在线,内容很全面!</b>"
msg[4]="<b>新浪网,去看看吧!</b>"
var browser;
if (document.all) {
        layerRef='document.all.'
        styleRef='.style.'
        changeMessages=".innerHTML=msg[num]"
        closeit=""
        browser=true}
else
{alert("此效果在 Netscape 浏览器中不能实现!");}
function mover(num){
    if(browser){
        eval(layerRef+'startingMsg'+changeMessages);
        eval(layerRef+'startingMsg'+closeit);}
}
function mout(num){
    if(browser){
        eval(layerRef+'startingMsg'+changeMessages);
        eval(layerRef+'startingMsg'+closeit);}
}
```

“解释菜单.js”文件要存放在包含它的.htm 文件相同的文件夹内。

2. 创建 CSS 文件

例 4.21 编写用来定义页面使用样式的 CSS 文件,保存文件为“解释菜单的样式.css”。

解释菜单的样式.css 文件的代码如下：

```
<style type="text/css">
<!--
#startingMsg{position:absolute; background:#FFFFFF; height:10; top:70; text-align:
middle;
  font-family:Arial;color:#000000; left:200}
  b{font-family:"宋体"; color:red;}
  a:link{color:green; text-decoration:none}
  a:visited{color:red; text-decoration:none}
  a:hover{color:blue; text-decoration:overline; font-size:26pt}
-->
</style>
```

“解释菜单的样式.css”文件存放在与包含它的.htm 文件同级的 CSS 文件夹内。例如，在“程序”文件夹下包含有 HTML 与 CSS 两个同级的子文件夹，HTML 中存放.htm 文件，CSS 子文件夹存放 CSS 文件。

当然可以存放在包含它的.htm 文件的相同文件夹内，只是在链接 CSS 文件是路径不同。

注意：CSS 文件与 JS 文件不同，它以<style type="text/css">标记码开头，以</style>标记码结束。

3. 创建包含 JS 与 CSS 文件的 HTML 文件

例 4.22 编写显示解释菜单的页面文件，保存文件为 4-20.htm。

4-20.htm 文件的代码如下：

```
<html>
<head>
<link rel=stylesheet href="../CSS/解释菜单的样式.css" type="text/css">
<script src="解释菜单.js" language="JavaScript" type="text/javascript">
</script>
</head>
<body>
<table width="760" border="0" cellspacing="0" cellpadding="0">
  <tr><td>
  <a href="4-1.htm" onMouseOver="mover(0)" onMouseOut="mout(0)">4-1.htm</a>
  <p><a href="4-2.htm" onMouseOver="mover(1)" onMouseOut="mout(1)">4-2.htm</a>
  <p><a href="4-3.htm" onMouseOver="mover(2)" onMouseOut="mout(2)">4-3.htm</a>
  <p><a href="http://www.263.net" onMouseOver="mover(3)" onMouseOut="mout(3)">首都在线</
a>
  <p><a href="http://www.sina.com" onMouseOver="mover(4)" onMouseOut="mout(4)">新浪网
</a>
</td></tr>
</table>
```

```
<div id="startingMsg" style="left:106px; top:72px; width:260; height:16">
这里可以显示不同的菜单解释！
</div>
</body>
</html>
```

在浏览器中打开 4-20.htm 网页，可以看到如图 4.21 所示的页面效果。

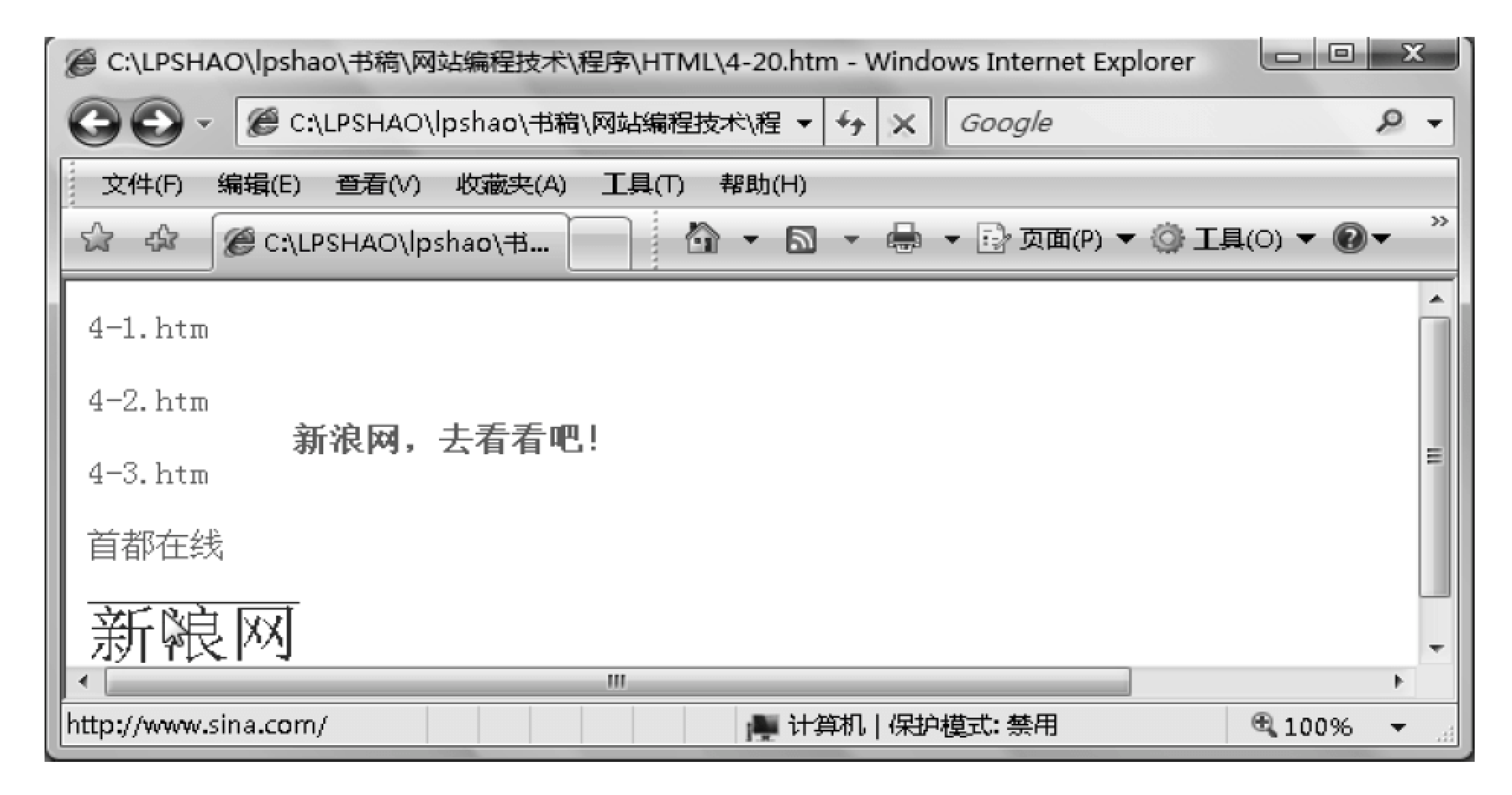


图 4.21　网页 4-20.htm 的页面效果

4.5.2　淡入淡出的图片

例 4.23　一个包含淡入淡出图片的文件(4-21.htm)。

```
<html>
<body>
<script language="JavaScript">
var b=1;
var c=true;
function fade(){
        if(document.all);
        if(c==true) {b++;}
        if(b==100) {b--; c=false}
        if(b==10) {b++; c=true;}
        if(c==false) {b--;}
  girl.filters.alpha.opacity=0+b;
  setTimeout("fade()",100);
  }
setTimeout("fade()",100);
</script>
<img src="img/3.gif" width="120" height="90" name="girl" alt="Image" style="filter:
alpha(opacity=0)">
```

```
<p style="position:absolute; left:206; top:40; width:200; height:100">淡入淡出的图片!</
p>
</body>
</html>
```

在浏览器打开 4-21.htm 网页，可以看到如图 4.22 所示的页面效果。



图 4.22　网页 4-21.htm 的页面效果

4.5.3　以不同方式轮显图片

例 4.24　一个可以不同方式轮流显示不同图片的文件(4-22.htm)。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>轮显的图片</title>
<script language="javascript">
var adImg=new Array();
var cID=0;
adImg[0]="img/1.jpg";
adImg[1]="img/2.jpg";
adImg[2]="img/3.jpg";
adImg[3]="img/4.jpg";
adImg[4]="img/5.jpg";
adImg[5]="img/6.jpg";
adImg[6]="img/7.jpg";
adImg[7]="img/8.jpg";
adImg[8]="img/9.jpg";

var preImgs=new Array();
for(i=0; i<9; i++)
    {
    preImgs[i]=new Image();
```

```
        preImgs[i].src=adImg[i];
        }
function NextAd()
        {
        if(cID==9)cID=0;
        if(document.all)
        {
        frmLS.imgLS.filters.revealTrans.Transition=Math.floor(Math.random() * 23);
        frmLS.imgLS.filters.revealTrans.apply();
        frmLS.imgLS.src=adImg[cID];
        frmLS.imgLS.filters.revealTrans.play();
            cID++;
        setTimeout("NextAd()", 10000);
        }
        }
</script>
</head>
<body background=img/xuehua2.gif>
<h1>轮流显示不同的图片效果!</h1>
<form name=frmLS>
<p align="center">
<img src="javascript:NextAd()" name="imgLS" width="550" height="400" hspace="0" vspace
="0" border="0" align="left"
style="FILTER:revealTrans(duration=6,transition=20); BORDER-RIGHT:# cccccc 1px solid;
BORDER-TOP:# cccccc 1px solid; BORDER-LEFT:# cccccc 1px solid; BORDER-BOTTOM:# cccccc 1px
solid" >
</p>
</form>
</body>
</html>
```

在浏览器打开 4-22.htm 网页，可以看到如图 4.23 所示的页面效果。

4.5.4 在网页中使用 CSS 样式的三种方式

通过本章的内容，可以看到在网页中使用 CSS 样式有如下 3 种不同的方式。

1. 引入

在 HTML 文档的<style>标记码对中引入 CSS 代码。
引入方式如下：

```
<style type="text/css">CSS 代码</style>
```

其中<style>中的"type='text/css'"的含义是告诉浏览器<style>标记码对中的代码是用来定义样式的。



图 4.23　网页 4-22.htm 的页面效果

2. 内嵌

即把 CSS 代码直接嵌入在 HTML 标记码行内，直接将 CSS 样式定义在指定标记码上，例如一个表格、一个图片或一段文字的标记码中，这个样式不会被其他标记码使用。

内嵌方式如下：

```
<标记码 style="属性名:属性值; 属性名:属性值; …; 属性名:属性值; ">…</标记码>
```

3. 链接

即把 CSS 代码单独保存为后缀名为.css 的文本文件，然后在使用样式的网页中引入 CSS 文件。

引入方式如下：

```
<link rel=stylesheet href="样式表文件名.css" type="text/css">
⋮
```

其中<link>标记的属性 rel＝stylesheet 用来说明链接的对象是一个样式表(stylesheet)文件，属性 href="样式表文件名.css"用来指定 CSS 文件的名称与路径。

链接这种方式非常适宜多个页面使用相同的样式，可减少大量的冗余代码。

例如，可以通过下面的代码使用 CSS 文件 manager.css 和 peixun.css 设置 HTML 文件的风格。

```
<head>
```

```
<link rel=stylesheet href="CSS 文档/manager.css" type="text/css">
<link rel=stylesheet href="CSS 文档/peixun.css" type="text/css">
</head>
```

4. 优先级

以上 3 种应用 CSS 的方式可在同一网页文件中使用，它们的优先级是内嵌＞引入＞链接，可见越靠近页面元素的样式属性优先级越高，即元素最终显示的样式是优先级最高的样式定义，同时各元素会自动继承上级元素中定义的样式。

4.6 思考与练习

4.6.1 思考题

4-1 什么是 CSS？它有什么作用？

4-2 定义 CSS 的语法结构有哪两种结构？

4-3 定义 CSS 有哪几种方式？在应用定义的样式时有什么不同？

4-4 如何将 CSS 定义代码引入到网页文件中？

4-5 如何使用 CSS 字体属性设置页面的字体格式？

4-6 如何使用 CSS 颜色和背景属性设置页面字体的颜色和背景效果？

4-7 如何为文本和图片设置边距属性？

4.6.2 上机练习

4-1 创建一个包含有文字样式、背景和颜色样式、空间定位样式的.css 文件，并将.css 文件引入.htm 文件中使用其设置的样式。

4-2 创建一个网页文件，使用 CSS 属性在打开网页时可以随机显示不同的图片。

4-3 创建一个包含文本与图片对象的网页文件，对图片设置透明效果，对文本设置透明效果。

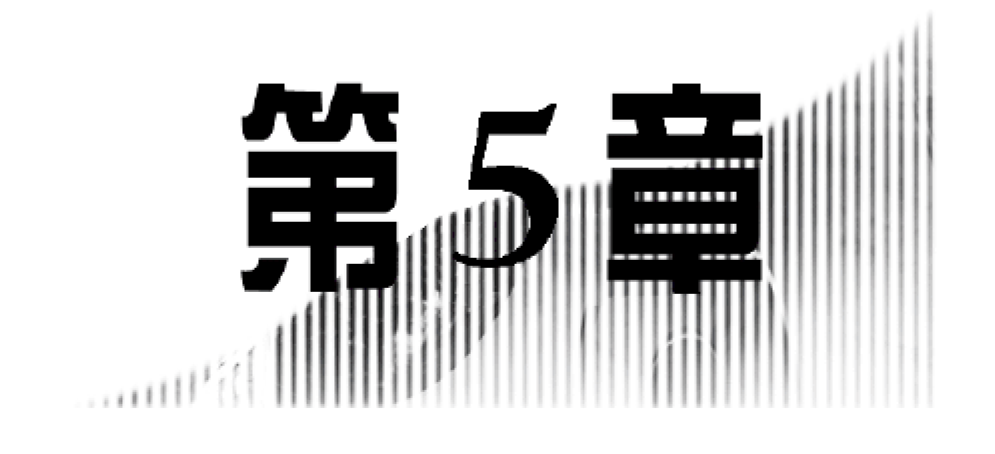
4-4 创建一个包含文本的网页文件，对文本设置模糊效果、投影效果。

4-5 创建一个包含文本与图片对象的网页文件，对文本和图片分别设置波纹效果。

4-6 创建一个包含文本的网页文件，对文本设置不同的发光效果。

4-7 创建一个包含图片对象的网页文件，对图片设置 X 射线效果、翻转效果。

4-8 分别创建 JS 文件、CSS 文件与.htm 文件，通过.htm 文件在页面指定位置显示一个淡入淡出的图片、在指定位置显示向上滚动的菜单，在指定位置显示带有边框的文字。



第5章 Java 服务网页——JSP

在 Sun 公司正式发布 JSP(Java Server Pages) 后,这种新的 Web 应用开发技术很快引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。按照 Sun 公司的说法,JSP 能够适应市场上包括 Apache Web Server 、IIS 在内的 85%的服务器产品。从本章开始将介绍如何使用 JSP 创建动态网页的方法。

本章的内容主要解决以下问题:

- 如何构建 JSP 的运行环境;
- 如何使用 JSP 创建动态网页;
- JSP 有哪些基本语法知识;
- JSP 有哪些常用的内置对象;
- JSP 有哪些常用的动作标记码;
- 什么是 JavaBean;
- 如何在 JSP 文件中使用 JavaBean;
- 如何通过虚拟目录运行 JSP 文件。

5.1 构建 JSP 的运行环境

本节主要介绍构建 JSP 运行环境的方法与步骤。

5.1.1 下载与安装 SDK

1. 下载 SDK 软件包

为了建立 JSP 的运行环境,需要先下载 Sun 公司免费的、最新的 Java EE SDK 软件包。Java EE SDK 是集成的开发环境,同时可下载捆绑的 JDK 软件包,JDK 中包含对编程最有用的是 Java 编译器、Applet 查看器和 Java 解释器。

下载软件 Java EE SDK 可参考以下步骤(注意,由于网站经常更新,页面会不同):

(1) 在浏览器中输入 http://java.sun.com/,可看到图 5.1 所示的 Sun 公司网站页面。



图 5.1　下载 Java EE 5 SDK

(2) 在页面右边“Popular Downloads”栏中可看到免费下载的软件名称，在 Java EE 5 SDK 超链接上单击打开如图 5.2 所示页面。

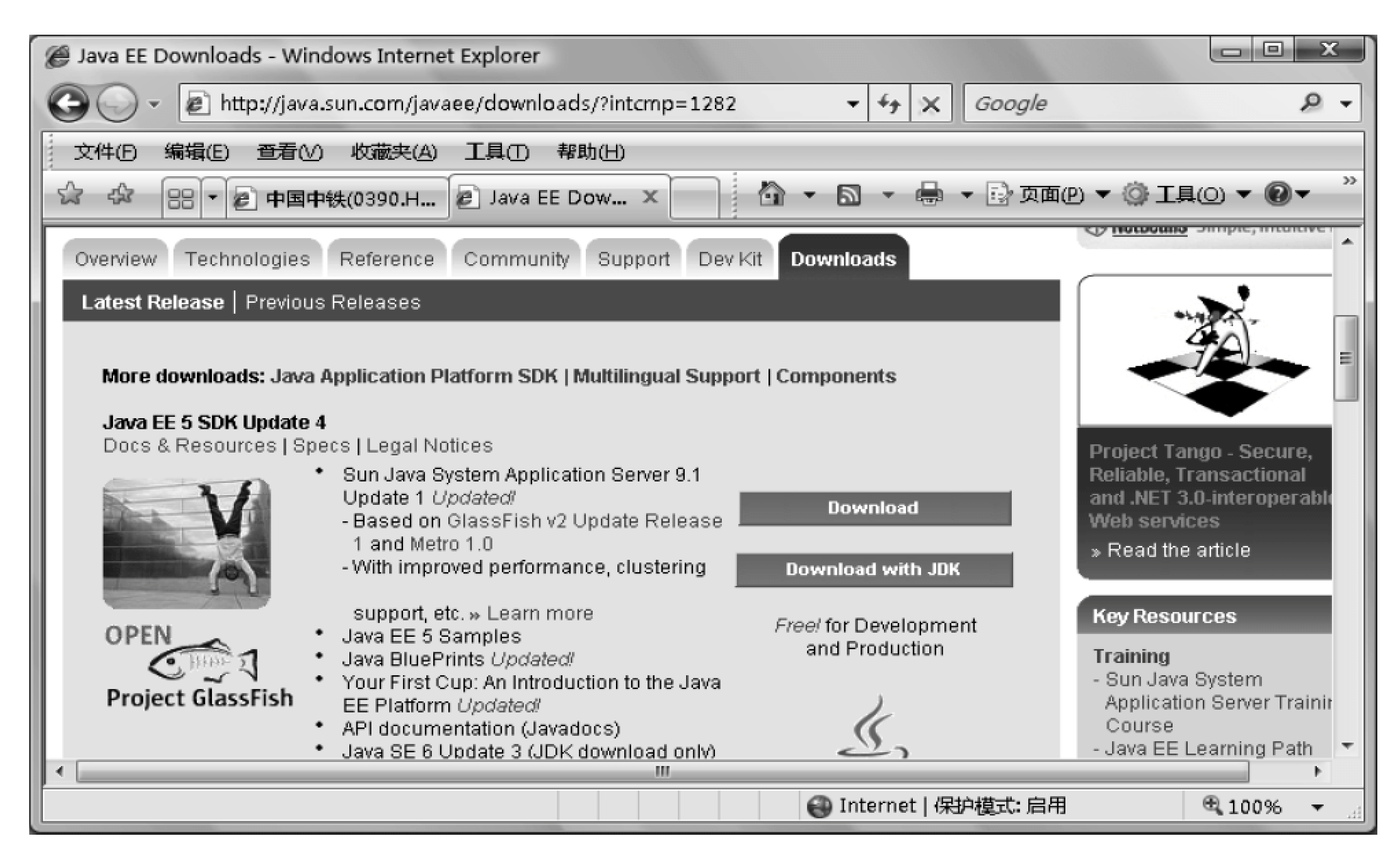


图 5.2　选择“包括 JDK 下载”

(3) 如图 5.2 所示，在页面 Downloads 标签下可看到免费下载的软件名称，这里显示的是“Java 应用程序平台工具包更新版(Java EE 5 SDK Update)”，在其栏中单击 Download with JDK 按钮，可打开图 5.3 所示的页面。

(4) 选择“I agree …”选项，同意 Sun 公司的协议，单击 Continue≫按钮，可进入到图 5.4 所示的下载页面。选择“Java EE 5 SDK Update 4 (with JDK)”选项。

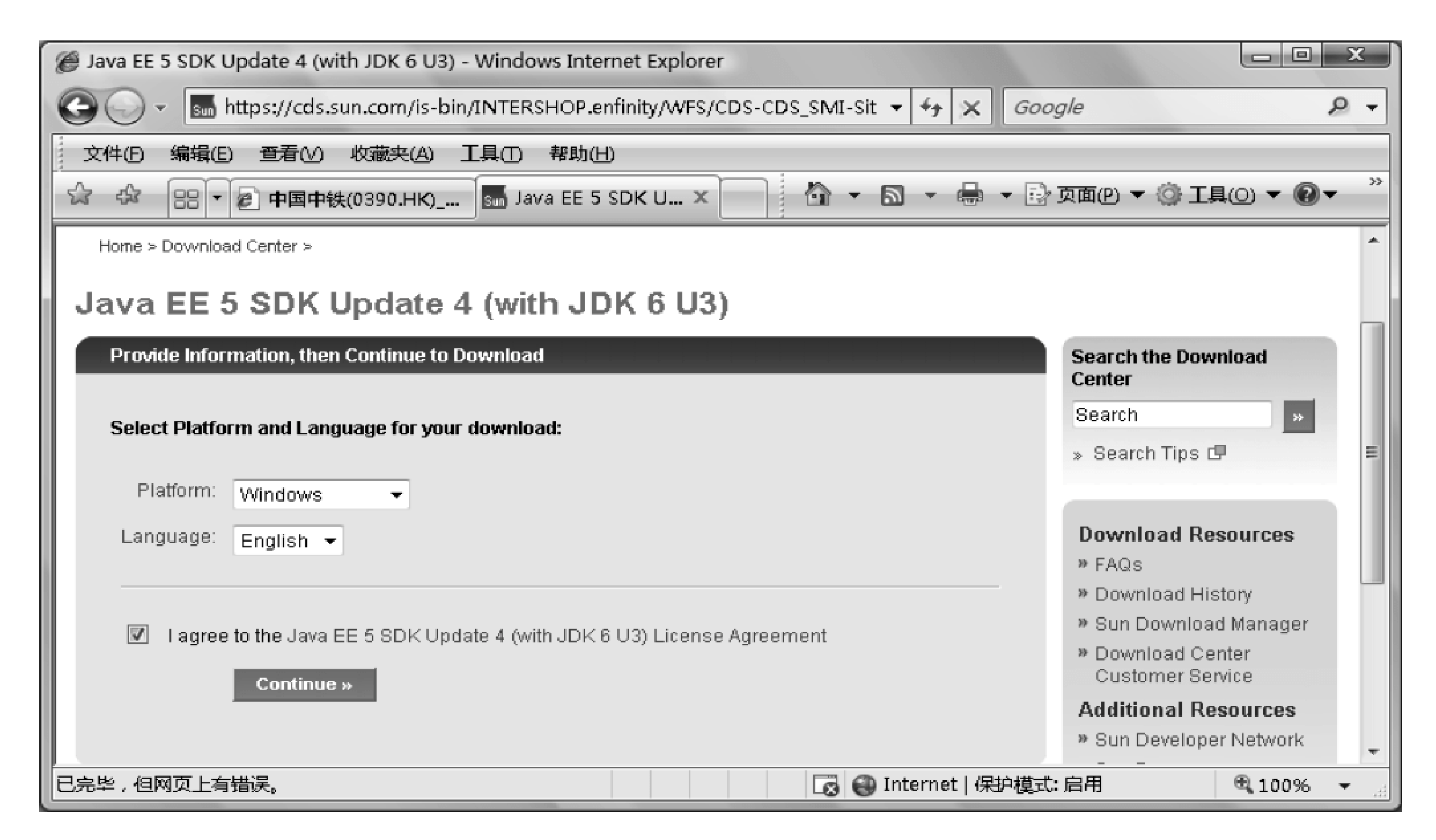


图 5.3　接受协议

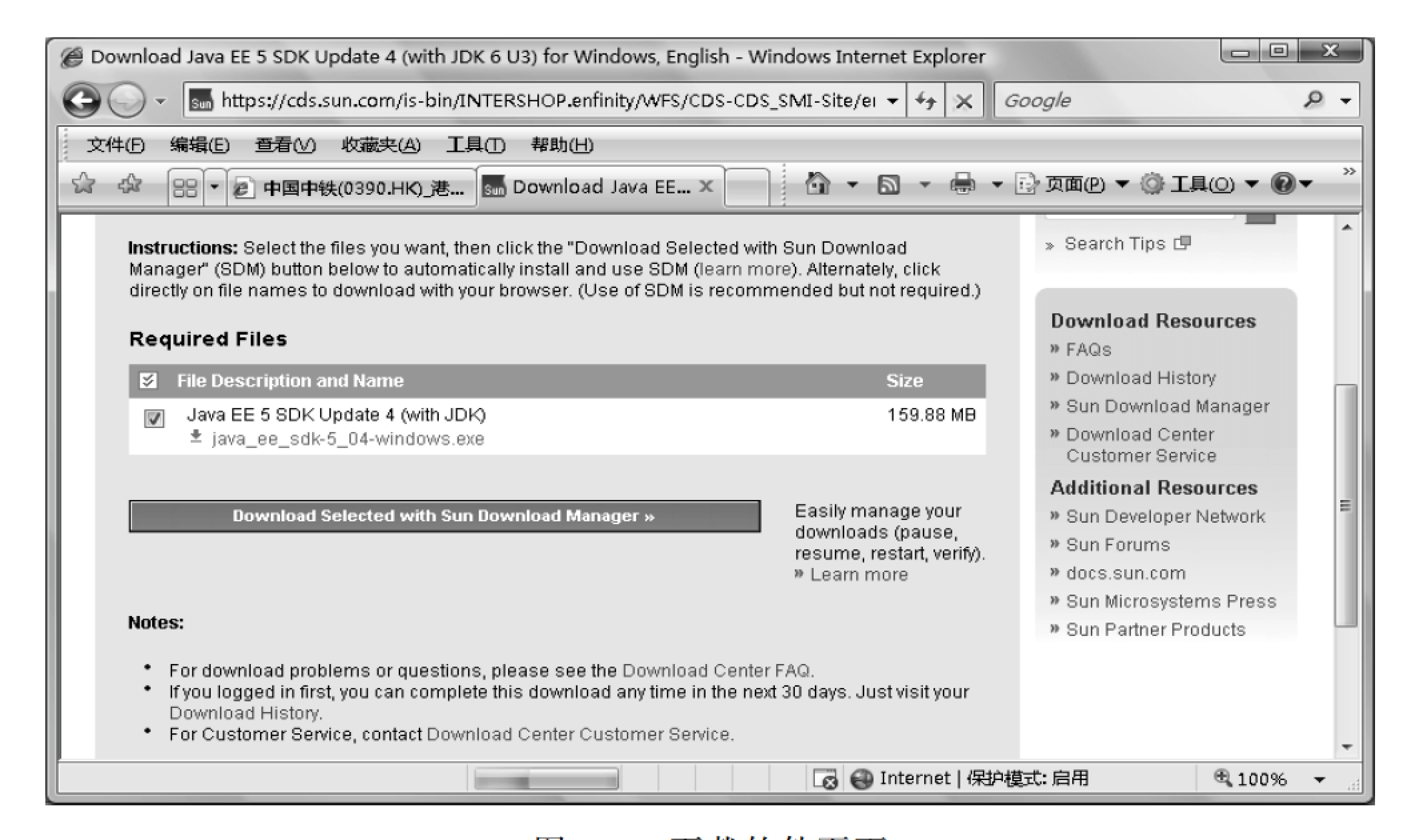


图 5.4　下载软件页面

(5) 单击 Download Selected with Sun Download Manager≫按钮，下载一个 Sun 提供的下载管理器，开始下载工作，如图 5.5 所示。

2. 安装 SDK

运行下载的"java_app_platform_sdk-5_04-windows.exe"软件包，可安装 SDK，在安装过程中可以设置安装路径及选择组件，系统默认的安装路径为 C:\sun\sdk（本书选择的安装路径为 C:\sdk）。

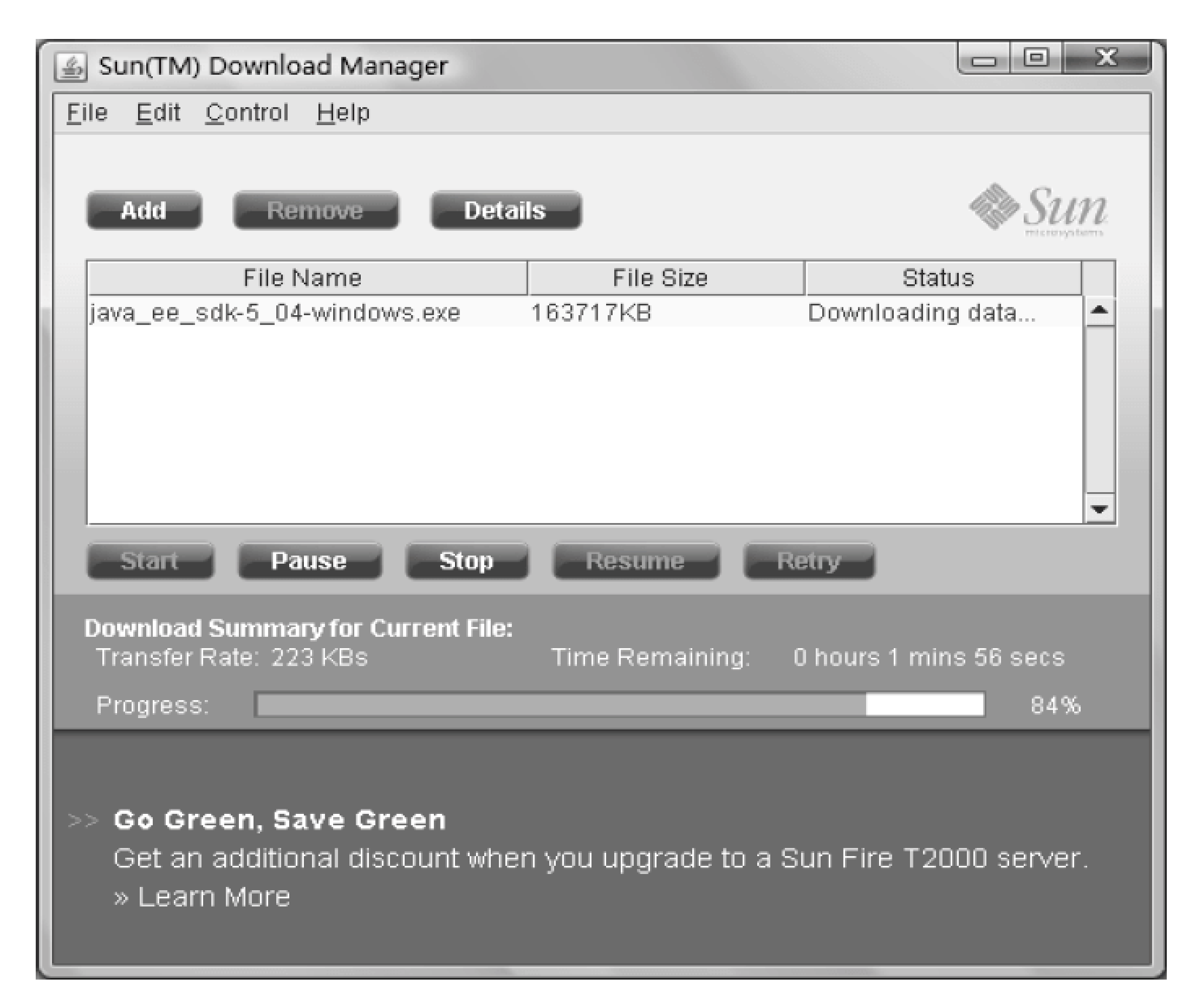


图 5.5　下载软件管理器

安装过程与其他软件基本相同，选择接受协议，单击 Next 按钮即可。在确定安装路径页面中，注意修改安装路径，例如，C:\sdk。

在确定用户名与口令安装页面中，可选择任意的用户名，例如 lpshao，口令为 8 位，例如，12345678。

5.1.2　下载安装 Java 软件

为了在浏览器中正常的运行 JSP，还要下载安装 Java 软件（Java Runtime Environment Version 5.0 Update 11），在 Sun 公司网站主页（http://java.sun.com/）可以看到 Get Java Software 图标，单击该图标可打开如图 5.6 所示页面，单击“立即下载”按钮或“手工下载”超链接，可下载该软件。

5.1.3　下载与安装 Tomcat

为了在服务器中管理与运行 JSP 文件，还需要下载 Tomcat 软件。Tomcat 是 Apache 组织的产品，是一个开放源代码的软件，它完全免费，文档齐全，配置容易，它是运行 servlet 和 JSP 的服务器管理软件，负责处理用户发送来的请求，并把该请求送到指定的 servlet，处理完之后再将结果送给用户。

1. 下载 Tomcat

下载 Tomcat 软件的可到 Apache 公司网站：http://tomcat.apache.org/，也可在国内站点免费下载。其 Apache 网站页面如图 5.7 所示。在其左边单击 Tomcat 6.x 超链接，打开下载 6.x 版本及更新版本页面，如图 5.8 所示。（建议下载 Tomcat 6.x 版本，本书以 6.0.14 版本为例介绍，下载时注意要选择支持 Windows 系统）。



图 5.6　Java 软件下载页面

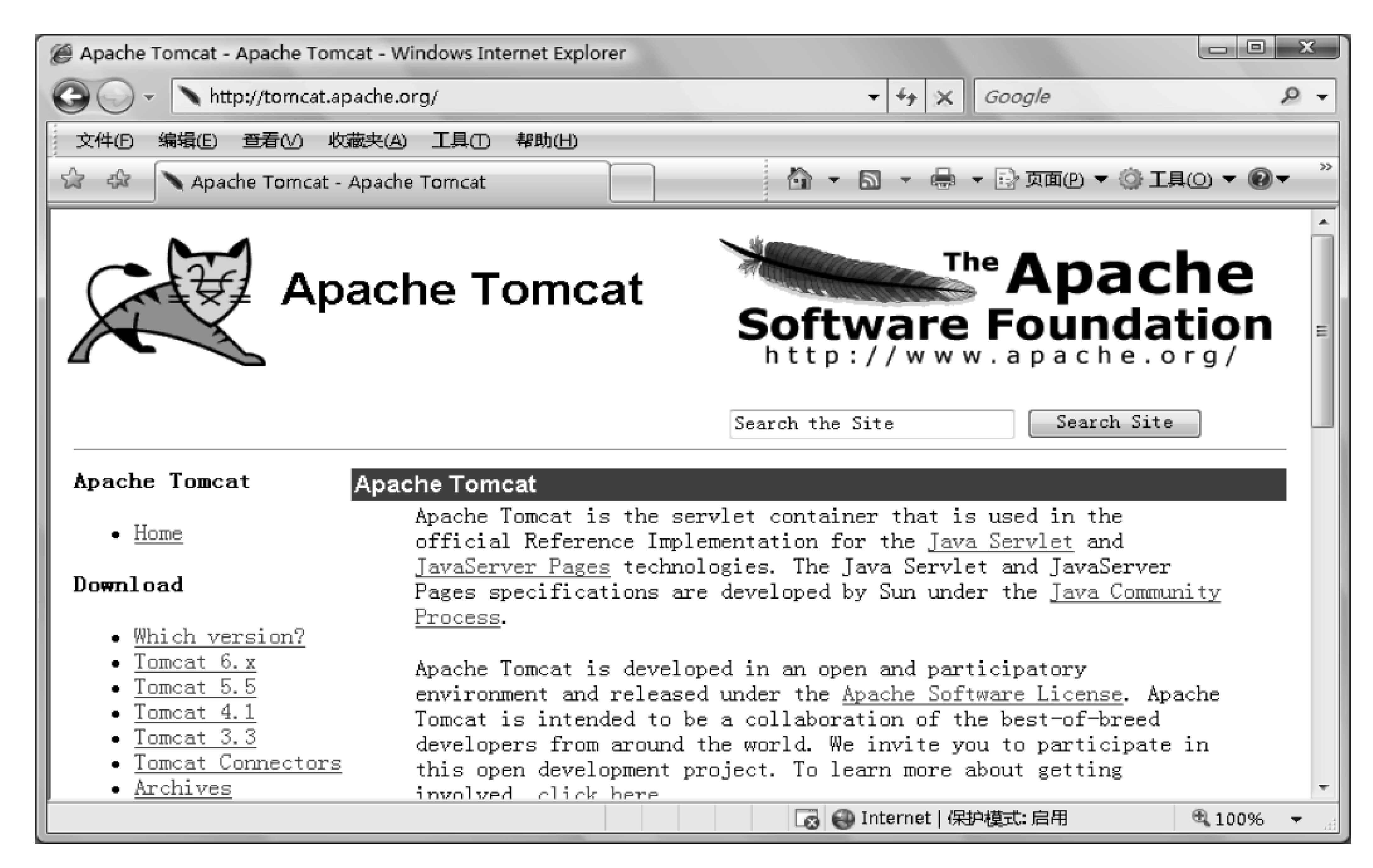


图 5.7　Apache 网站主页

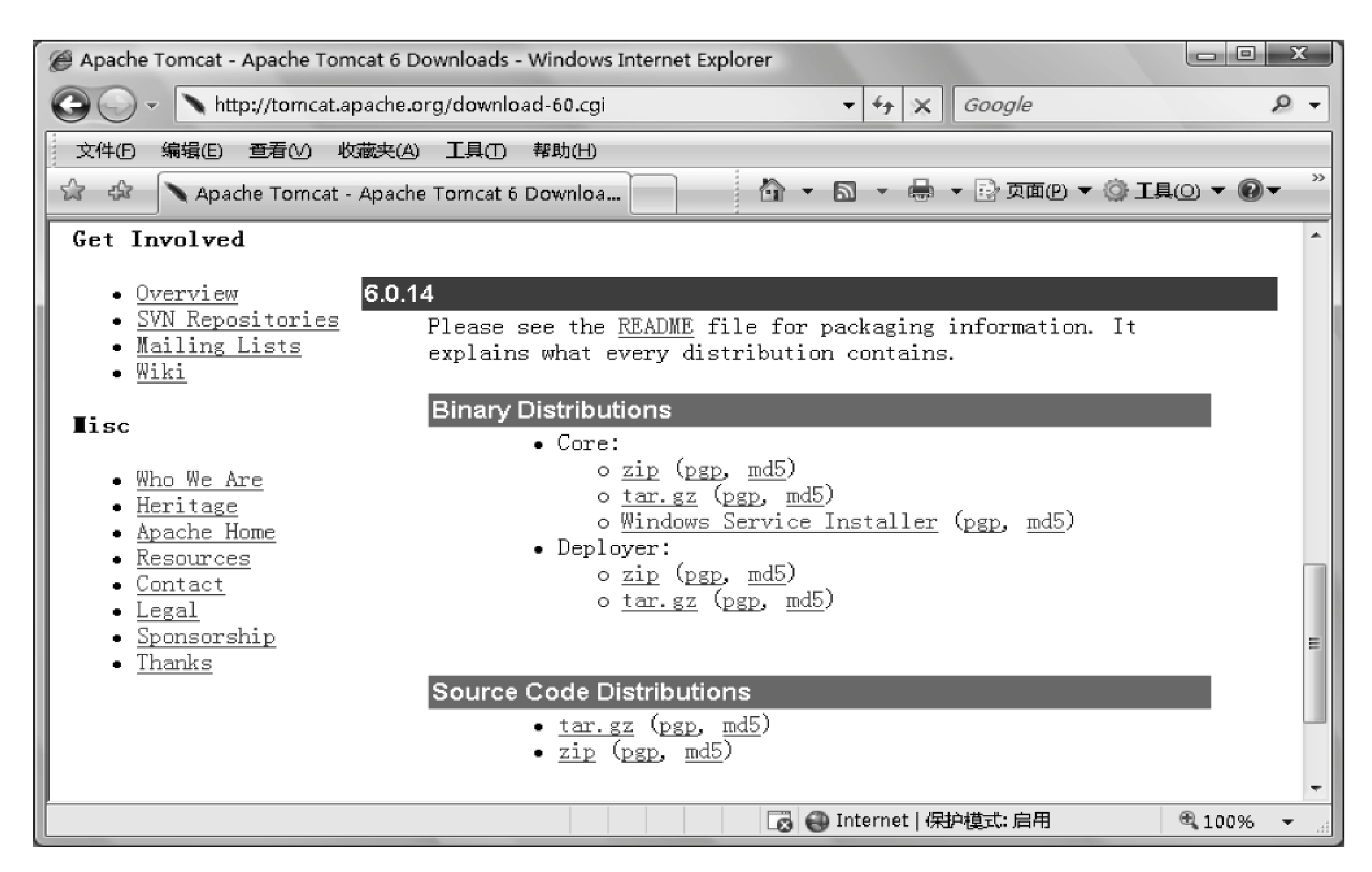


图 5.8　Tomcat 6.x 下载路径

2. 安装 Tomcat

安装 tomcat 可参考如下步骤：

(1) tomcat 下载完成后，再其存放目录双击 jakarta-tomcat-6.0.14.exe，可开始安装。显示如图 5.9 所示的安装页面。



图 5.9　欢迎安装的对话框页面

(2) 在欢迎安装对话框界面单击 Next 按钮显示如图 5.10 所示的协议页面。

(3) 在如图 5.10 所示协议对话框中，单击 I Agree 按钮，显示如图 5.11 所示页面，选择 Full 完全安装方式，选择 Examples 选项，单击 Next 按钮显示如图 5.12 所示页面。

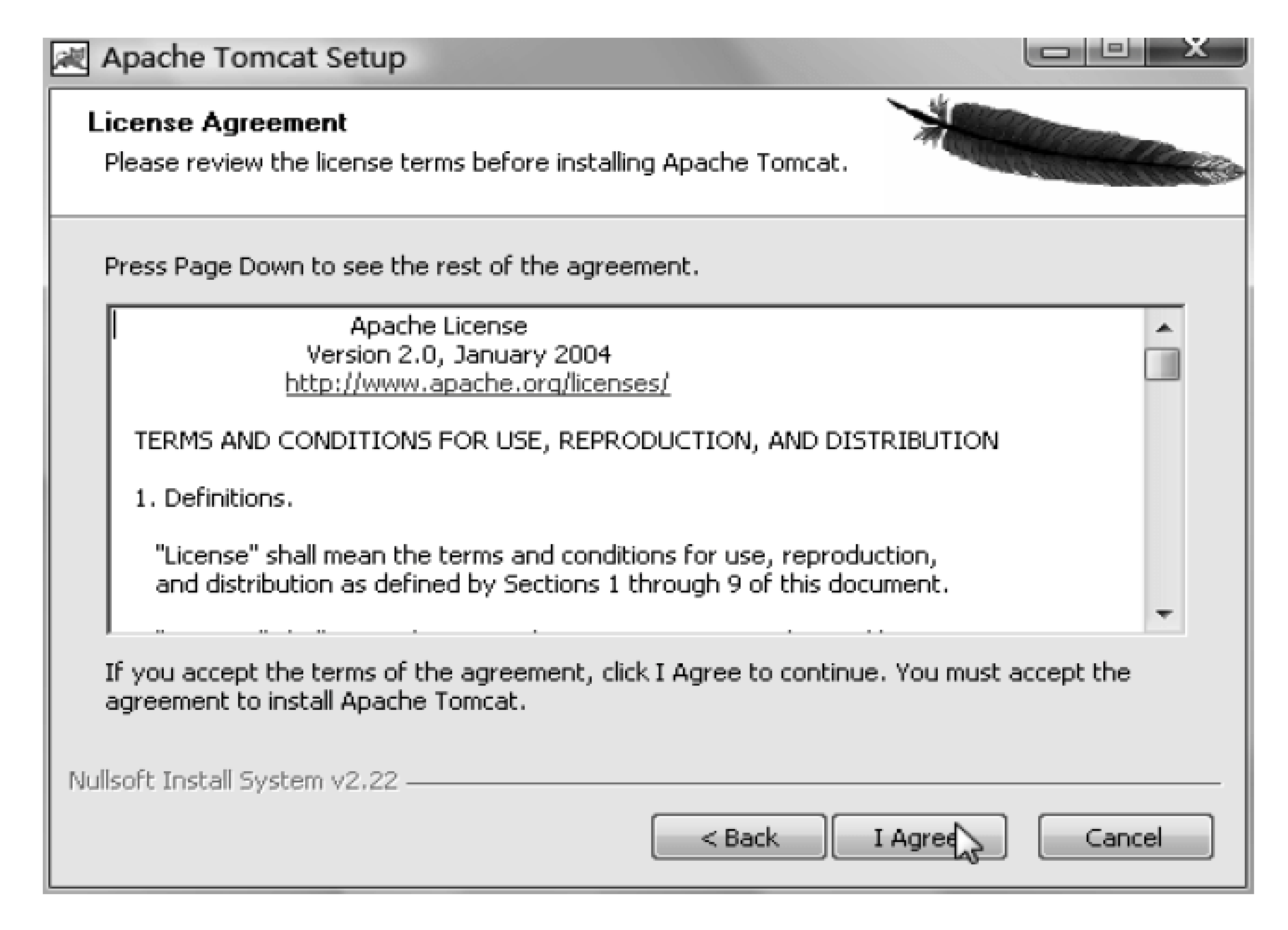


图 5.10　协议对话框页面

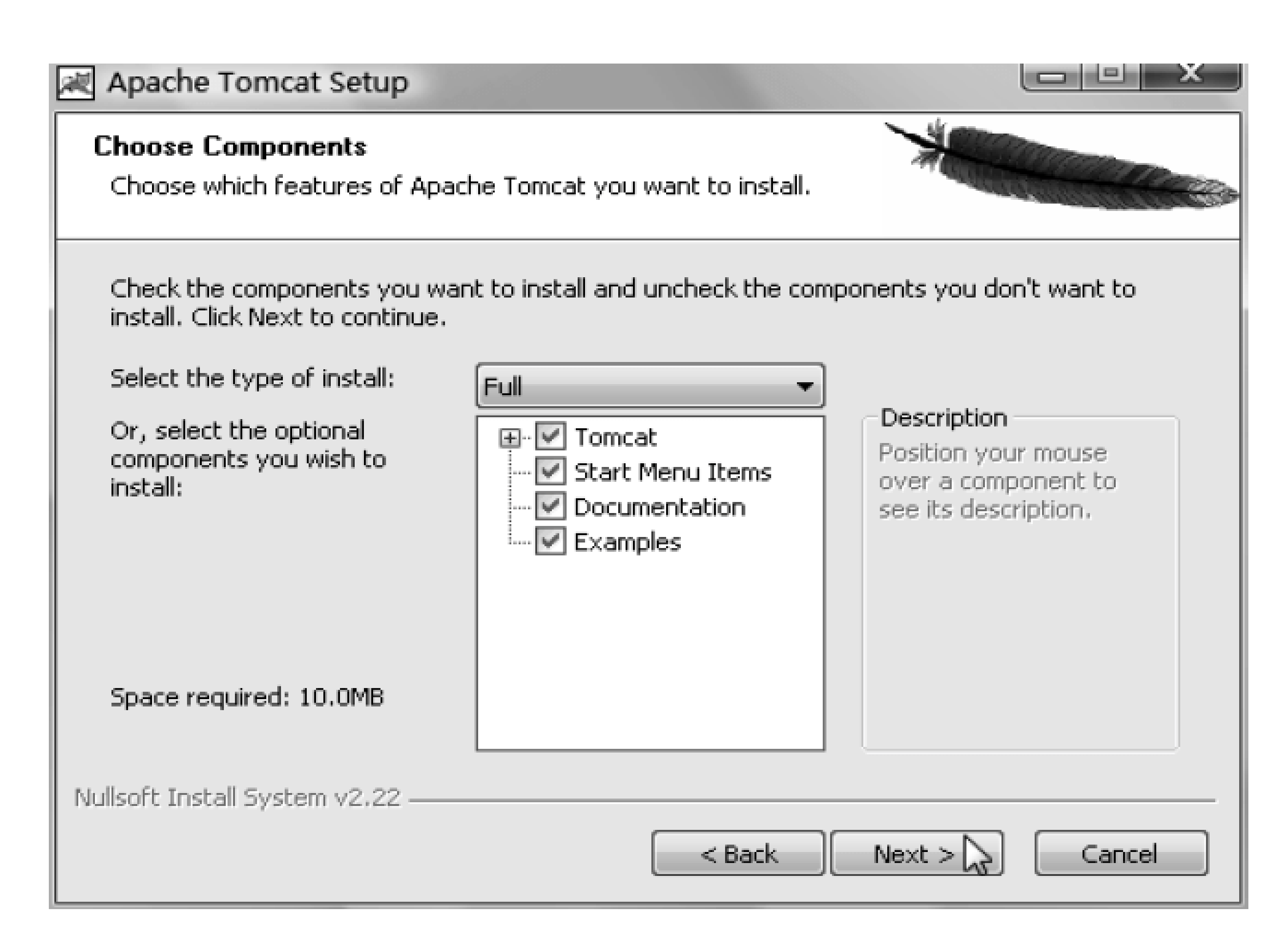


图 5.11　选择安装方式

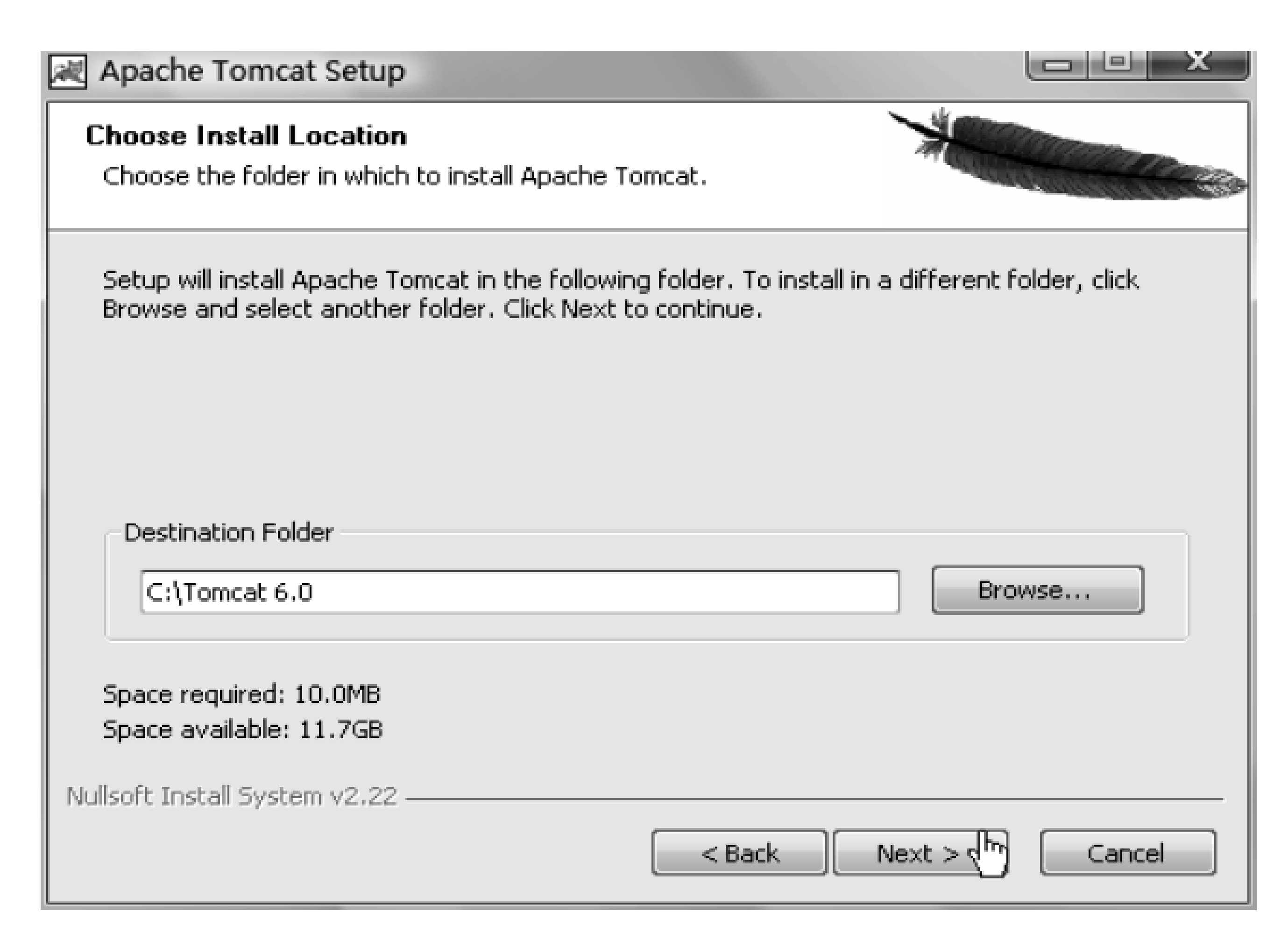


图 5.12　选择安装路径

(4) 在如图 5.12 所示页面中，选择 Tomcat 软件的安装路径（本书选择 C:\Tomcat 6.0），单击 Next 按钮显示如图 5.13 所示页面。

(5) 在如图 5.13 所示对话框中，修改用户名，本书修改为 lpshao，口令为空，单击 Next 按钮显示如图 5.14 所示页面。

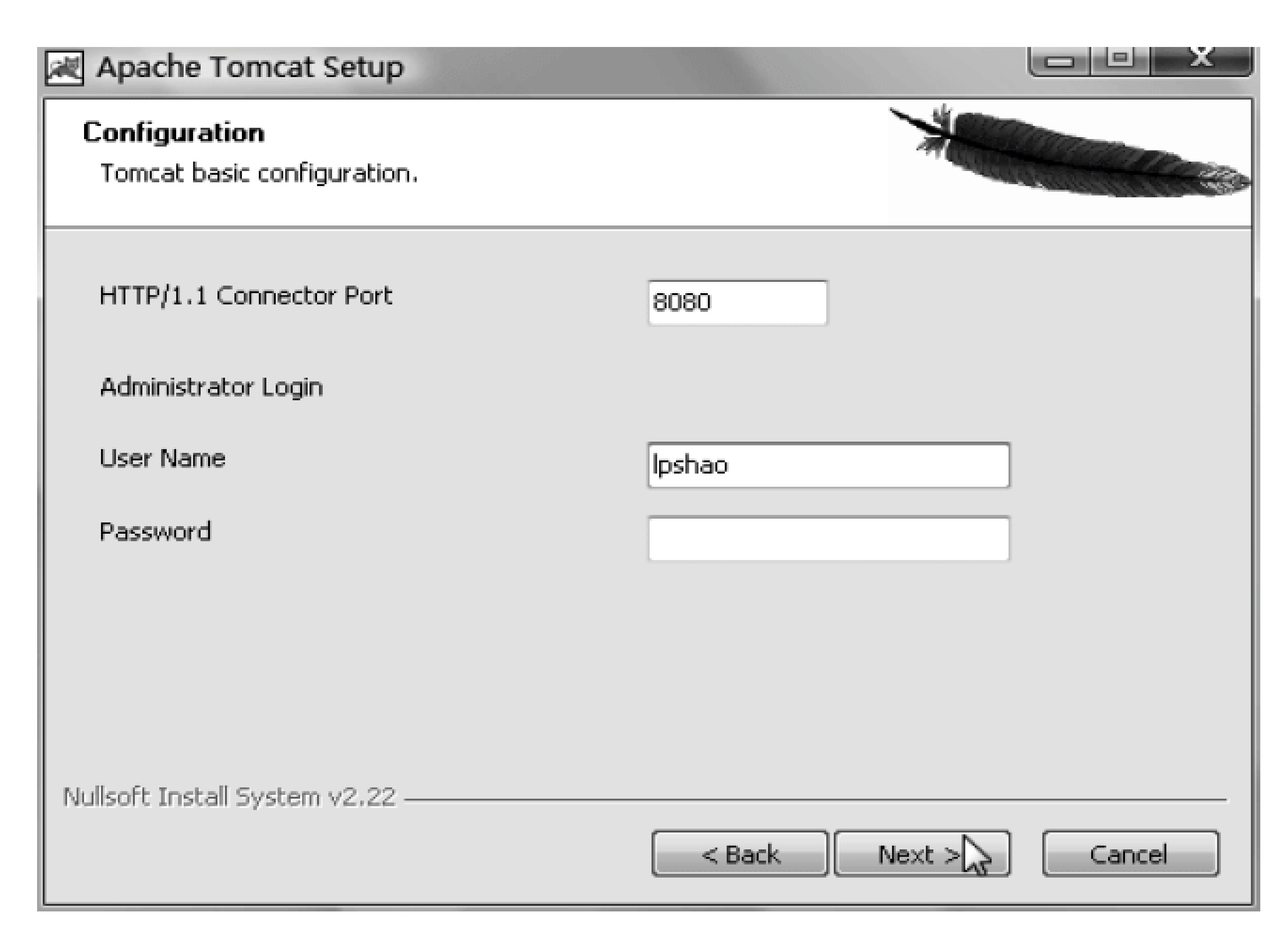


图 5.13　选择用户名与口令

(6) 选择 Java 虚拟机的路径，输入 c:\sdk\jdk（或系统自找的 Java 虚拟机路径），单击 Install 按钮，开始在计算机中安装 Tomcat 软件。

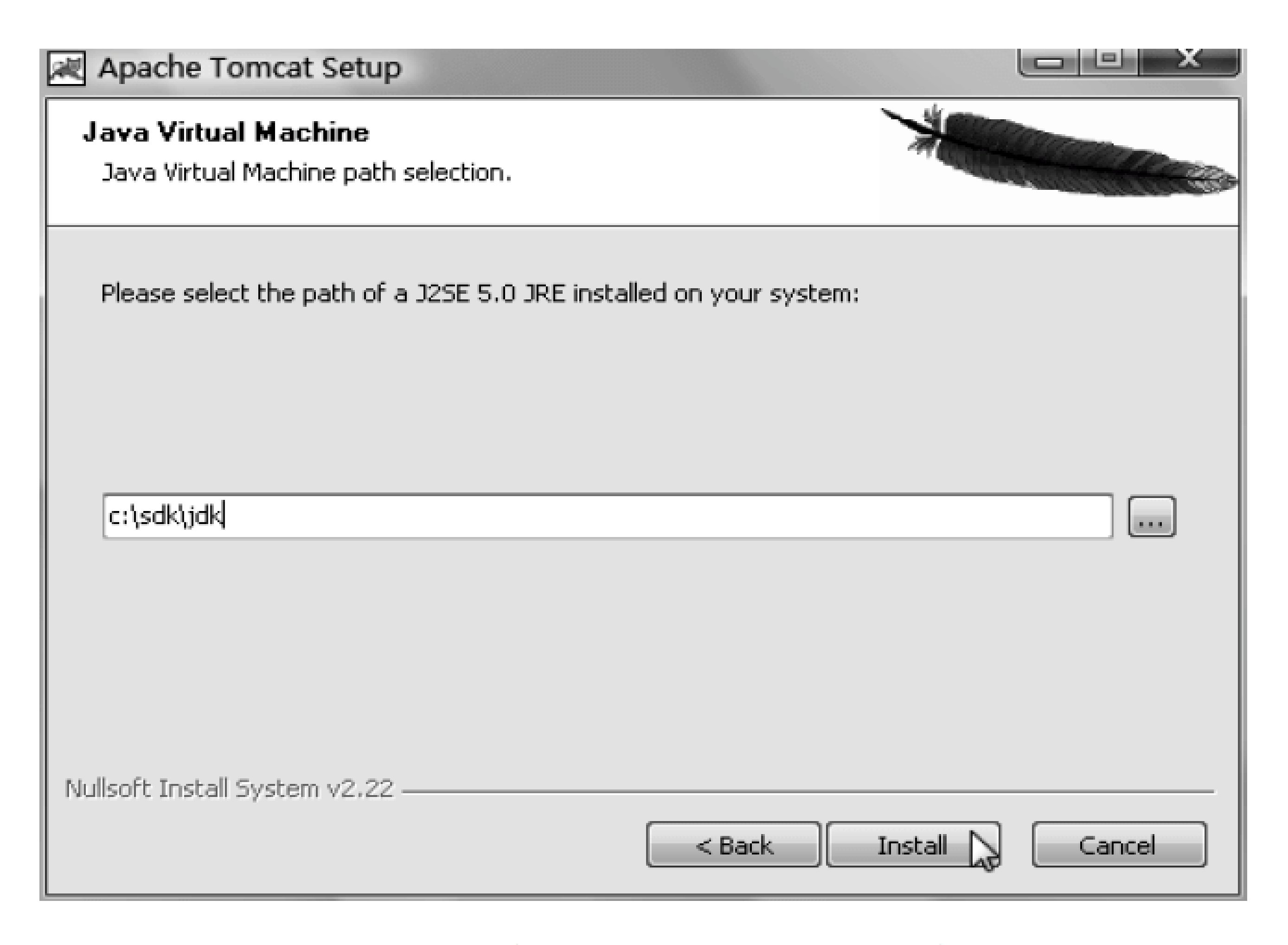


图 5.14　确定 Java 虚拟机的路径

(7) 耐心等待系统自动安装完毕，会出现安装进度显示，如图 5.15 所示。

(8) 安装完毕后，会显示如图 5.16 所示界面，单击 Finish 按钮可结束 Tomcat 软件安装工作。

(9) 接着会在“记事本”打开英文的使用注意事项。在计算机屏幕右下角会看到 Tomcat 服务器图标。在浏览器中输入 http://localhost:8080，如果看到如图 5.17 所示的 tomcat 欢迎页面，说明 Tomcat 安装成功了。

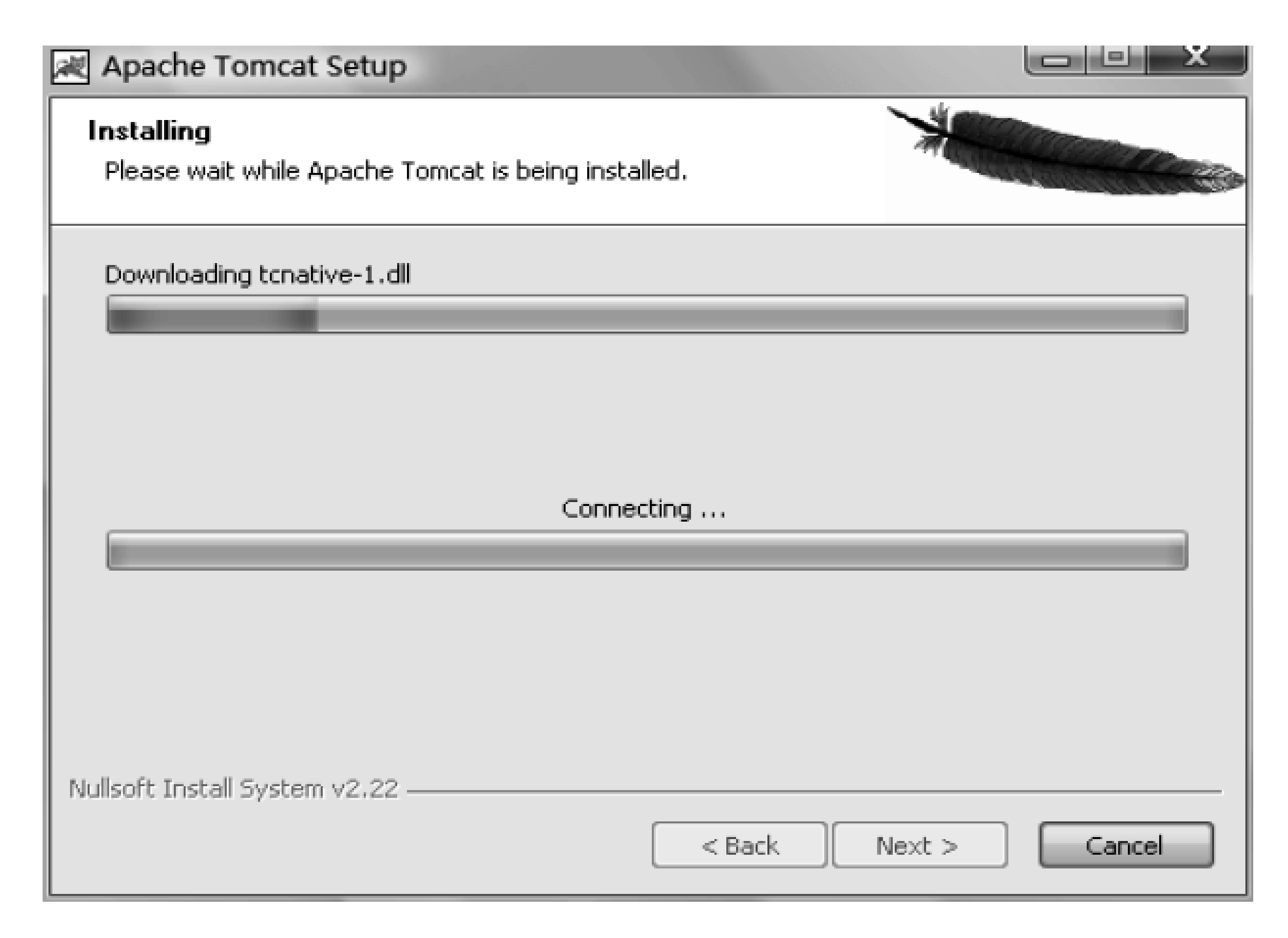


图 5.15　安装状态

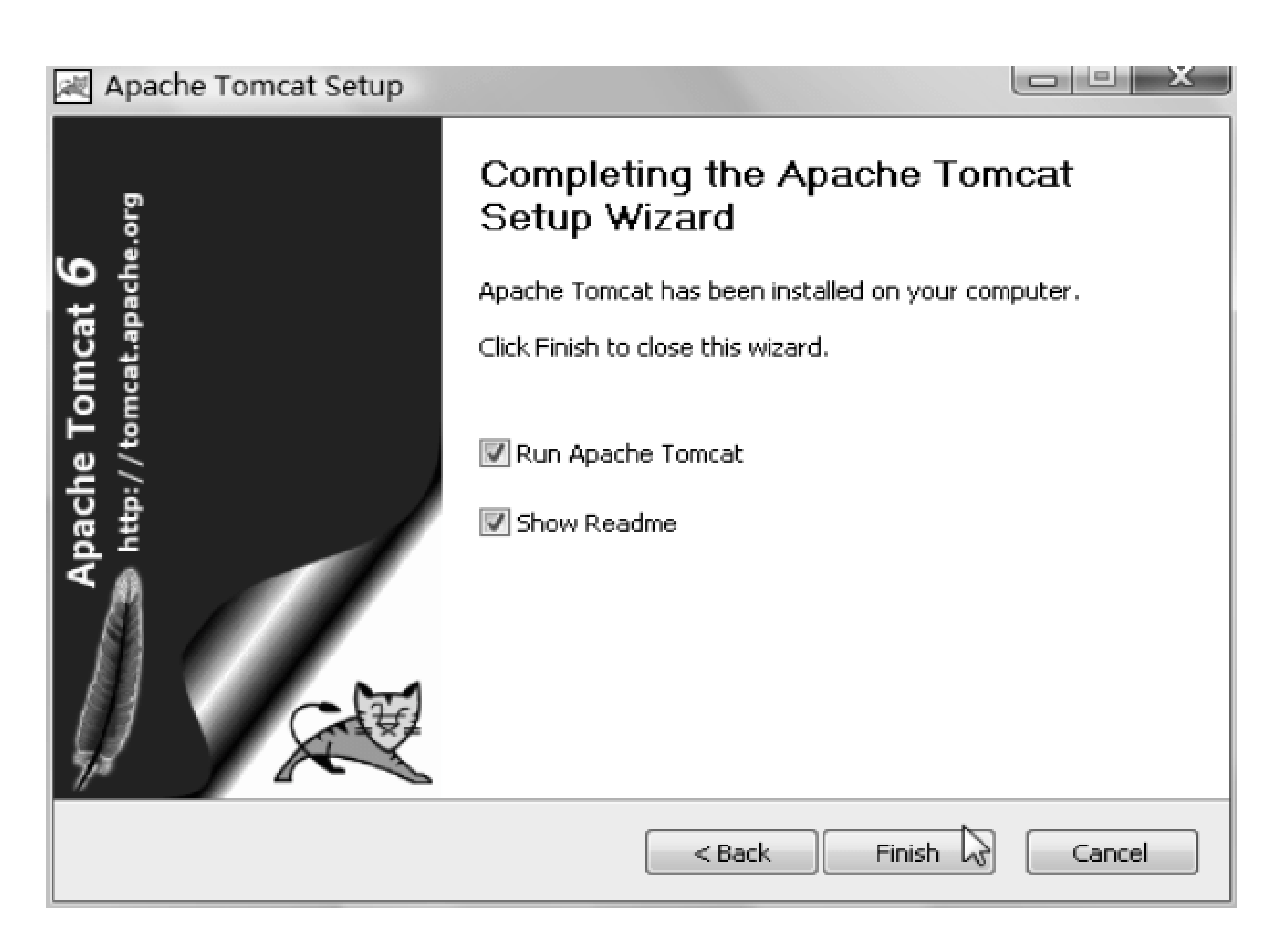


图 5.16　安装结束页面

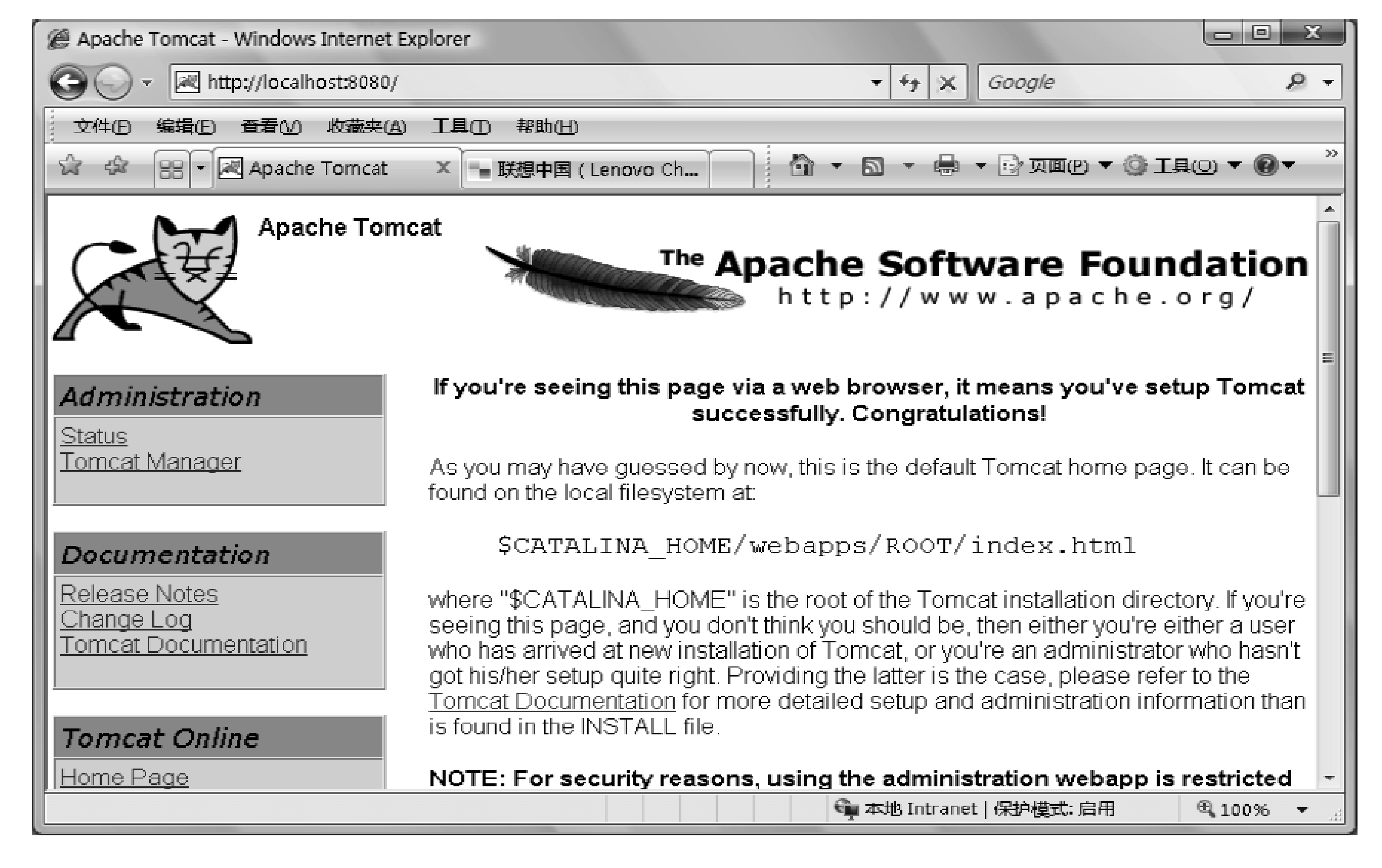


图 5.17　Tomcat 欢迎页面

(10) 启动与关闭 Tomcat 服务器

在 Windows 界面选择“开始”→“所有程序”→Apache Tomcat 6.0→Monitor Tomcat 或 Start Tomcat 命令，在计算机屏幕下方会看到 Tomcat 开始运行的图标，如果图标为，可右击图标在快捷菜单选择 Start Service 命令，服务器启动后右击图标在快捷菜单选择 Stop Service 命令将停止 Tomcat 服务，选择 Exit 可关闭 Tomcat 服务器。

3. 设置 Windows 的环境变量

使用版本高的 Tomcat(6.0.X 以上)中可以不设置环境变量，即可在打开 Tomcat 服务器时正常运行 JSP 文件。

如果 Tomcat 版本低，不能正常运行 JSP 文件，可参照下面的介绍去设置 Windows 系统的“环境变量”。

设置 Windows“环境变量”的操作步骤：

(1) 在 Windows 桌面上右击“计算机”(我的电脑)→“属性”命令，打开“系统”页面如图 5.18 所示，单击“高级系统设置”选项，打开“系统属性”对话框，从中选择“高级”标签，如图 5.19 所示。



图 5.18 系统页面

(2) 在图 5.19 中单击“环境变量”按钮，打开“环境变量”对话框，如图 5.20 所示。

(3) 在“用户变量”框中单击“新建”按钮，分别创建环境变量：JAVA_HOME、TOMCAT_HOME、PATH、CLASSPATH。SDK 的安装目录在 JAVA_HOME 变量之中，Tomcat 的安装目录在 TOMCAT_HOME 变量之中。

- JAVA_HOME 变量值：C:\sdk\jdk(要看自己 SDK 的安装路径)。
- TOMCAT_HOME 变量值：C:\Tomcat 6.0(要看自己 Tomcat 的安装目录)。
- CLASSPATH 变量值：“.;%JAVA_HOME%\bin;%TOMCAT_HOME%\lib;”或“.;%JAVA_HOME%\bin;%TOMCAT_HOME%\lib;%TOMCAT_HOME%\lib\jsp-api.jar; %TOMCAT_HOME%\lib\servlet-api.jar;c:\sdk\



图 5.19 “系统属性”对话框



图 5.20 “环境变量”对话框

jdk\lib\tools.jar”。

• PATH 变量值：C:\sdk\jdk\bin。

（4）单击“确定”按钮，可结束环境变量的设置。

5.1.4 创建 JSP 用户目录

在 C:\Tomcat 6.0\webapps\ROOT 根文件夹下，创建一个用户使用的文件夹

myjsp,以后编写的 JSP 程序将放在 myjsp 目录下。

5.1.5 运行 JSP 程序

1. 编写第一个 JSP 网页文件

例 5.1 一个显示当前日期和“世界你好”的 JSP 网页文件(5-1.jsp)。

```
<html>
<body>
<font color="blue">现在时间是
<%=(new java.util.Date()).toString()%>
<p>世界你好!
</font>
</body>
</html>
```

在浏览器中输入 http://localhost:8080/myjsp/5-1.jsp,运行结果如图 5.21 所示。

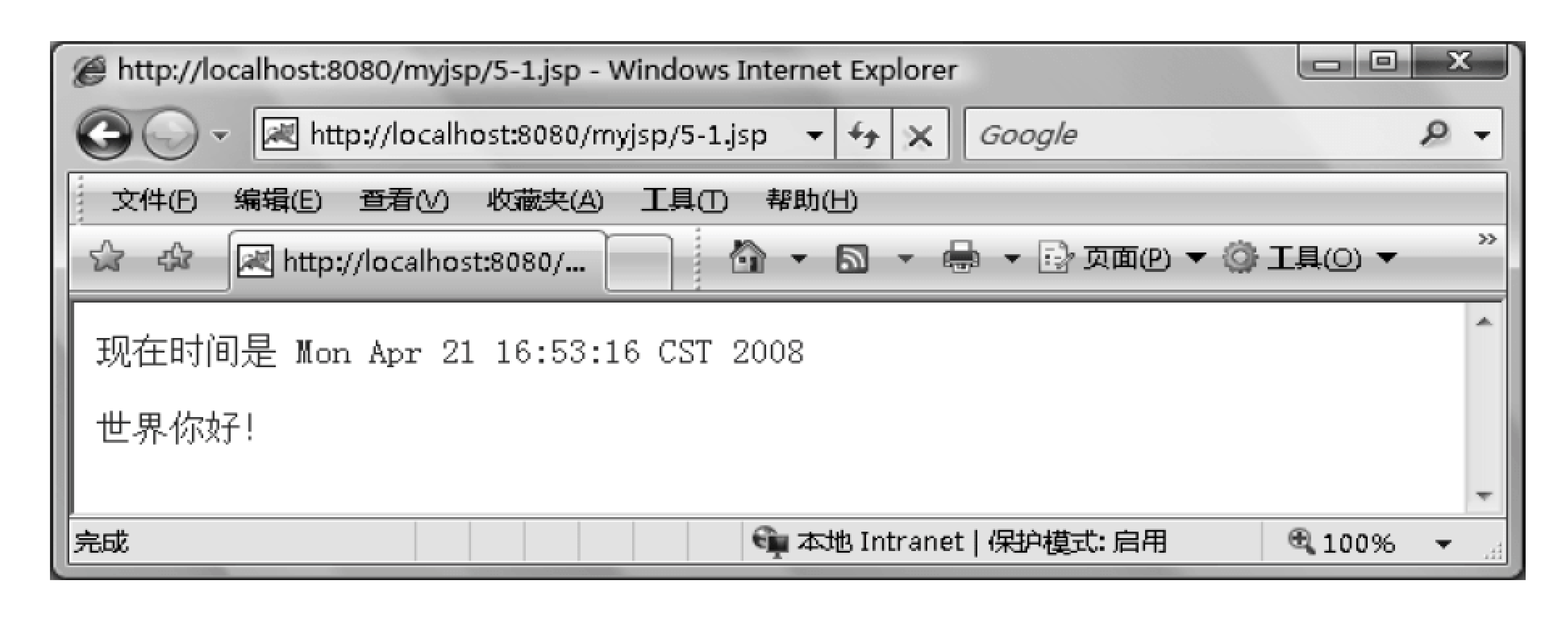


图 5.21 5-1.jsp 的运行结果

2. 知识点归纳

(1) 从例 5.1 可以看出,JSP 网页文档中使用的代码与 HTML 文档中的标记码几乎完全一样,只有<%=(new java.util.Date()).toString()%>这一行没见过。不过可以很容易猜出它的意思,无非是将当前日期显示出来。说明 JSP 与 HTML 有很多类似的地方。

(2) 同 HTML 文件比较,JSP 文件中多了两个符号<% 和 %>,这是 JSP 的定界符,它用来分隔 HTML 标记码与 JSP 代码,<% 和 %>中间的内容是 JSP 代码。

(3) 通过例 5.1 可知,JSP 文档主要包括 HTML 的标记码和 JSP 代码。HTML 标记码用一对尖括号<>括起来。JSP 的代码使用一对尖括号和百分号<%%>括起来。包含 JSP 代码的文件后缀名为.jsp,文件要存放在服务器下的\Tomcat 6.0\webapps\ROOT\myjsp 目录下。

5.1.6 编辑 JSP 文件的软件工具

在编写 JSP 网页文件时可使用 EditPlus 编辑软件，它可以编辑 HTML、ASP、JSP、Perl、Java 等语句，而且可以高亮显示语句区；还可以自己定义自己的工具栏，甚至快捷方式。使用起来很方便。在网上可以免费下载该软件。

在进行大型 JSP 项目时，还可以使用开发平台 MyEclipse，先到 Eclipse 官方网站 http://www.eclipse.org/下载 Eclipse 的安装软件，然后在 Eclipse 中安装 MyEclipse 插件即可组建一个 MyEclipse 开发平台。

5.1.7 JSP 的运行流程

JSP 具有一次编译，多次、多处执行，代码执行效率高的特点。当 Web 服务器第一次接收到用户的 JSP 页面请求时，JSP 容器(Tomcat)先将 JSP 页面编译为 Servlet 类文件(.class)，接着运行 Servlet 类文件对客户端的请求进行处理，执行其中的 Java 程序片段，然后将执行结果返回给 JSP 容器，最后以 HTML 格式传回给用户，如图 5.22 所示。

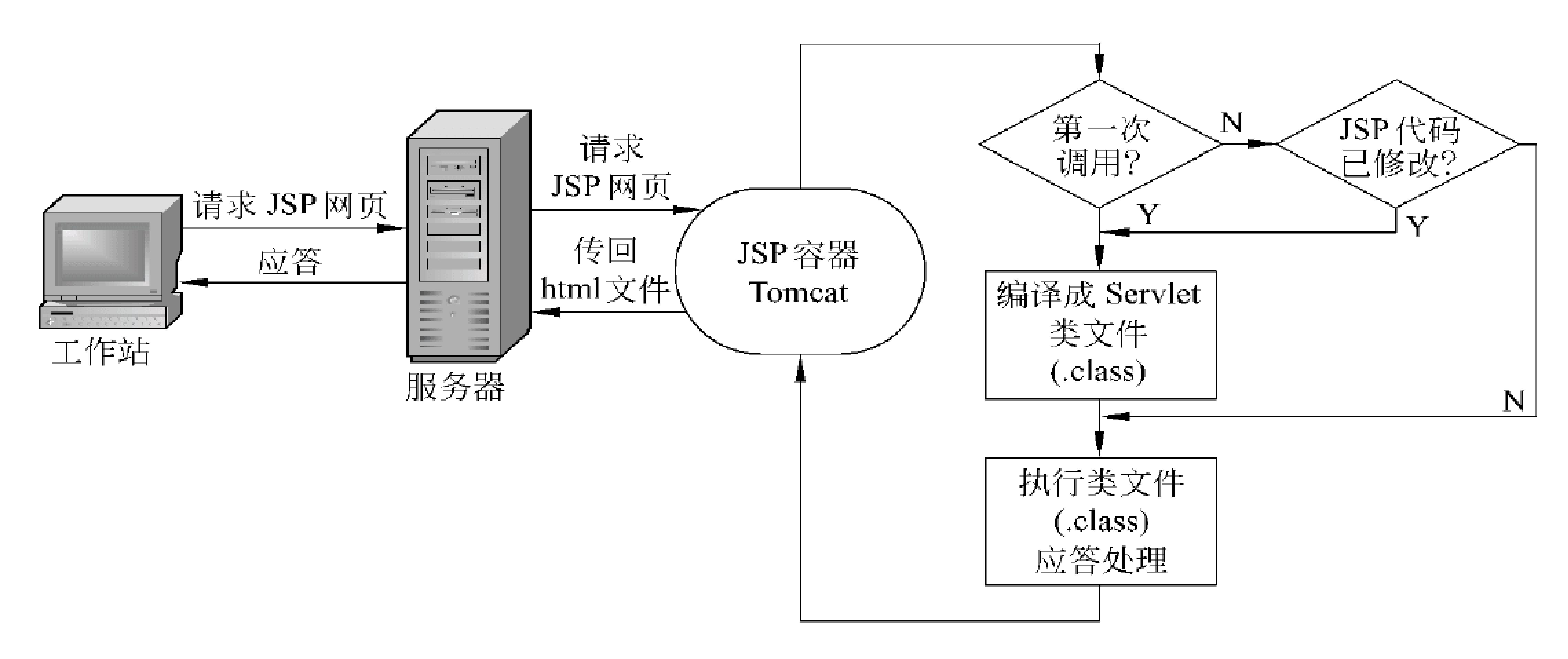


图 5.22 JSP 网页运行流程

以后再有同样的 JSP 请求时，无需重新编译，直接执行已编译好的.class 文件即可，因此执行效率特别高。如果 JSP 网页代码发生了改变，则需要重新编译，如图 5.22 所示。

5.2 JSP 的基本元素

JSP 网页就是在 HTML 网页中包含了能够产生动态内容的 Java 程序代码。在通常情况下，JSP 文件(以.jsp 作为后缀名)可以包含 HTML 语言编写的静态页面内容与 JSP 基本语句，JSP 基本元素主要包含 JSP 指令语句、JSP 注释语句、JSP 声明语句、JSP 表达式语句和 Java 程序片段。

本节的内容主要介绍 JSP 基本语句的使用方法。

5.2.1 JSP 指令语句

位于＜％@ ％＞标记中的代码，称为 JSP 编译器指示指令，简称为指令。JSP 指令用来设置整个 JSP 页面的相关属性，包括网页的编码方式、语言等。

JSP 指令语句的一般语法格式为：

```
<%@ 指令名 属性="属性值"%>
```

常用的指令有 page、include 和 taglib。

1. page 指令

page 指令用来定义 JSP 文件的全局属性，作用于整个 JSP 页面，如使用的语言、是否维持会话状态、是否使用缓冲区等。例如：

- 语句＜％@ page language＝"java" ％＞定义了本 JSP 文件使用的是 Java 语言。
- 语句＜％@ page import＝"java. util. Date" ％＞用来导入支持的 Java 类。
- 语句＜％@ page errorPage＝"errorPage. jsp" ％＞用来指定本 JSP 文件出现 Java 问题时显示的出错页面。
- 语句＜％@ page session＝"true" ％＞用来定义是否需要为用户管理通话期的信息 session。

2. include 指令

include 指令用来在 JSP 文件中包含一个静态文件，同时解析这个文件中的 JSP 语句。例如，语句＜％@ include file＝"filename. jsp" ％＞中使用 include 指令向 JSP 容器发送了一个消息：将在文件 filename. jsp 中找到的文本在转换后插入到当前的 JSP 网页中。

使用 include 指令即在当前文件中插入一个包含文件 filename. jsp，所包含的文件可以是固定格式的 HTML 或是 JSP 等。include 指令的优点是代码重用。

3. taglib 指令

taglib 是与标记（标签）相关的指令，用来定义一个标记库以及其自定义标记的前缀，声明此 JSP 文件使用的自定义标记。例如，通过下面的语句定义了自定义标记 loop。public 为自定义标记的前缀，uri 声明了标记库（命名空间）。

```
<%@ taglib uri="http://www.jspcentral.com/tags" prefix="public"%>
<public:loop>
 ⋮
</public:loop>
```

5.2.2 JSP 注释语句

JSP 注释语句是 JSP 文件中的注解语句，是为了增加程序的可读性而编写的语句，对

程序的运行结果没有影响。

JSP 注释语句格式如下：

```
<%--注释内容--%>
```

例如，下面的语句就是一个注释语句：

```
<%--这是 JSP 的注释语句--%>
```

注意：HTML 使用的注释标记为<!--这里是注解--!>；JavaScript 使用的注释标记为//（单行注释）和/ * ,…, * /（多行注释）。

5.2.3 JSP 声明语句

JSP 声明语句是用来在 JSP 文件中定义页面级变量或方法的。目的是让编译器知道哪些变量和方法是合法的。通过 JSP 声明语句可以声明一个或多个变量和方法，在声明语句中声明的变量和方法将在 JSP 页面初始化时被初始化。

JSP 声明语句的语法格式如下：

```
<%! 类型 变量或方法名; %>
```

例如，下面的声明语句声明了整型变量 i，字符串型变量 s：

```
<%! int i=0; %>
<%! String s="你好!"; %>
```

注意：%和!之间不能有空格，而且一条声明语句不能分在两行写，一定要以分号(;)结束声明语句，就像在普通 Java 中声明成员变量一样。因为 JSP 中的任何内容都必须是有效的 Java 语句。

在 JSP 中除了可以声明在 JavaScript 中使用的 6 种数据类型变量外，可以声明 Java 语言中支持的所有数据类型变量。

5.2.4 JSP 表达式语句

JSP 表达式语句提供了一种简单的输出形式，在运行后会将 JSP 生成的数值、转化的字符串嵌入在 HTML 页面的相应位置显示。其中的表达式与 JavaScript 中使用的表达式基本相同。

JSP 表达式语句的语法格式为：

```
<%=表达式%>
```

例如在例 5.1 中，使用了表达式输出了当前日期的值：

```
<%=(new java.util.Date()).toString()%>
```

还可以使用下面的语句输出已声明的整型数据 i 和字符串 s 的值：

```
<%=i%>
<%=s%>
```

5.2.5 Java 代码片段

1. Java 代码片段的作用

位于<% %>标记中的代码块用来嵌入描述 JSP 页面动态处理逻辑的 Java 语言代码,它们称为 Java 代码片段或脚本片段(Scriptlets)。Java 代码片段在 Web 服务器响应请求时会被服务器运行处理,完成动态处理功能。

2. 编写带有 Java 代码片段的 JSP 文件

例 5.2 使用 Java 的 for 循环语句片段以及 JSP 表达式分别以不同字体标记显示“你好”的 JSP 文件(5-2.jsp)。

```
<%--JSP基本语法练习--%>
<% for (int i=1; i<=4; i++){%>
<H<%=i%>>你好</H<%=i%>>
<%}%>
```

在浏览器中输入 http://localhost:8080/myjsp/5-2.jsp,运行结果如图 5.23 所示。

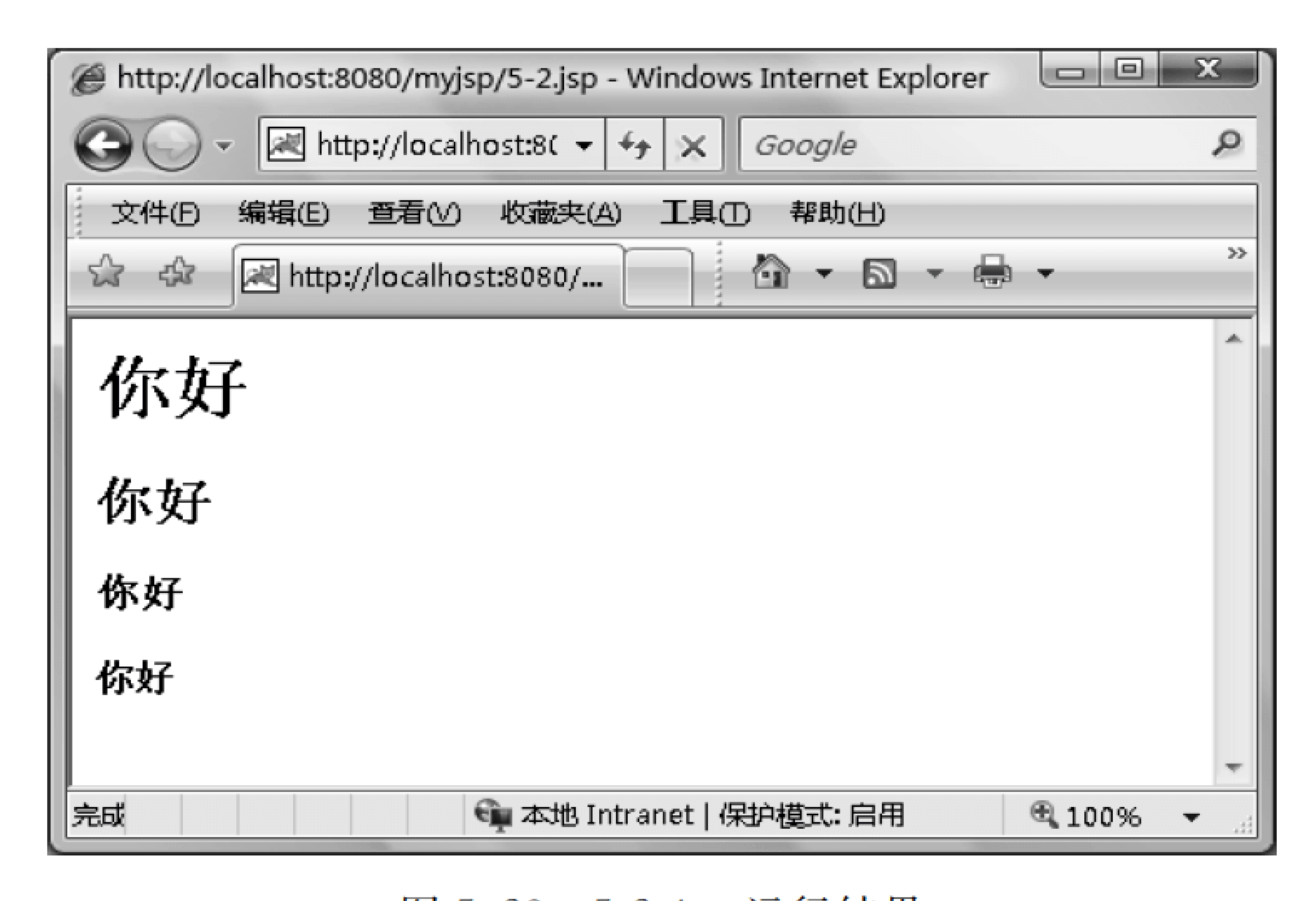


图 5.23 5-2.jsp 运行结果

3. 知识点归纳

(1) 在 JSP 文件中通过<% %>标记嵌入 Java 代码,例 5.2 中嵌入了“for(int i=1; i<=4; i++) {”。Java 的 for 循环语句。

(2) 在 JSP 文件中使用变量时需要要先声明变量类型与变量名。

(3) JSP 代码与 HTML 标记可以混合使用,在 HTML 标记中可以嵌入 JSP 代码。例如:

```
<H<%=i%>>你好</H<%=i%>>。
```

5.3 JSP 常用的内置对象

为了简化 JSP 语句，JSP 定义了一组可以直接使用的内置对象。这些对象包含在 JSP 页面中，浏览器在编译 JSP 页面时会自动识别所包含的内置对象。内置的意思是这些对象可以直接引用，不需要显式地声明，也不需要专门的代码创建实例对象。

本节的内容主要介绍几个常用的 JSP 内置对象的使用方法。

5.3.1 获取输入信息的 request 对象

下面通过一个 HTML 文件与一个 JSP 文件介绍如何使用 JSP 的内置对象 request 获得用户在页面上输入的信息并将信息在另一页面上显示。

1. 编写静态的 HTML 页面文件

例 5.3 创建一个用来输入用户信息的静态页面文件(5-3.html 文件)。

```
<html>
<head>
<title>输入信息页面</title>
</head>
<form method=post action="5-4.jsp">
输入姓名<input type="text" name="yhm">
<input type="submit" value="提交" >
</form>
</html>
```

2. 编写使用内置对象 request 的 JSP 文件

例 5.4 创建输出客户信息的页面(5-4.jsp)。

```
<font color="blue">欢迎你到这里来,
<%String yhm1=request.getParameter("yhm");%>
<%=yhm1%>
<p>这里是学习 JSP 动态网页编程的地方
</font>
```

在浏览器中输入 http://localhost:8080/myjsp/5-3.htm，运行结果如图 5.24 所示。

单击“提交”按钮后，会出现如图 5.25 所示页面，它是 5-4.jsp 运行的结果。

3. 知识点归纳

1) request 对象

5-4.jsp 中使用了 JSP 内置对象 request，它是 javax.servlet.ServletRequest 类的子



图 5.24 用户输入信息的页面



图 5.25 5-4.jsp 运行的结果

类对象，在传输协议 HTTP 下的 Web 应用中，request 对象作为 HttpServletRequest 类的一个对象，其作用是在 C/S 模式下，在客户端或服务器程序中获取发送的各类信息，以及 Web 服务器的参数。

可以说，request 对象为 jsp 程序提供了数据处理的“原材料”，没有 request 对象的话，jsp 程序就失去了与客户端联系的交互能力，可以把 request 对象看做 JSP 的眼睛和耳朵。

2）request 对象的生命期

request 对象只是在本次请求中有效，当信息发送到后，request 对象也将消失。

3）request 对象的常用方法

```
String getParameter(String name)            //得到客户端提交的指定属性 name 的值
String[] getParameterNames ()               //返回客户端向服务器端提交的所有参数
                                            //的名字，为 Enumeration 对象
String getServletPath ()                    //得到 Web 服务器中的路径
String getServletName ()                    //得到 Web 服务器名称
Int getServletPart ()                       //得到 Web 服务器端口号
String getQqureyStringPart ()               //得到 get 方法提交的数据
String setAttribute (String name,Object o)   //设置属性 name 的值
```

```
String getAttribute (String name)    //得到属性 name 的值,如果属性值不存在,返回 null
String[] getAttributeNames ()     //返回 request 对象所有属性的名字,为一个字符串数组
Cookie[] getCookies ()            //返回客户端的 Cookie 对象,结果为 Cookie 数组
String getMethod ()               //返回客户端向服务器提交数据的方法名
String getRemoteHost ()           //返回客户端的机器名
String getRemoteAddr ()           //返回客户端的 IP 地址
HttpSession getSession ()         //返回当前请求的 Session,如果不存在,新创建一个
```

4）使用 form 表单传递数据到 JSP 文件的方式

通过使用 request 对象,不仅可以获取从 HTML 文件中用 post 方法或 get 方法提交的数据,还可以访问任何用 HTTP 请求传递的信息。post 方法提交的数据是隐式的,常用来提交信息量较大、安全性高的信息,例如用户的基本信息。get 方法传递的数据是显式的,在地址栏中可以看到,一般用来提交信息量较少、安全性不高的信息,例如查询字符串。

使用 form 表单传递信息一般有 3 种方式:

(1) 由 HTML 网页内的 form 传递信息给其他 JSP 文件。

(2) 由其他 JSP 文件内的 form 传递信息给另外一个 JSP 文件。

(3) 由 JSP 文件内的 form 传送信息给自身(默认设置)。例如可在 test.jsp 中由 form 传递数据给 test.jsp 文件自身使用,可使用下面的语句:

```
<form method="post" action="test.jsp">
```

5）form 表单标记码及其属性

```
<form action=被调用程序的 url 网址 enctype=数据传送的 mime 类型 name=form 表单名称
method=提交数据的方式 get/post onrest=按下 rest 键所调用的程序 onsubmit=按下 sumit 键所
调用的程序 target=输出信息的窗口或网页的名称>
</form>
```

5.3.2 发送响应信息的 response 对象

response 对象与 request 对象类似,它是 java.servlet.ServletResponse 类的子类对象。在传输协议 HTTP 下的 JSP/Servle 应用中,response 对象作为 HttpServletResponse 类的一个对象,其作用是在 C/S 模式下,发送各种响应信息。

response 对象的常用方法如下:

```
void sendError(int code, String errorMessage)
                                   //服务器不能响应请求时,向客户端发送出错信息
```

例如,下面的语句可以用于当 Web 服务器不能响应用户请求时,使用状态代码向客户端返回的发现错误代码及出错信息:

```
<%response.sendError (403, "非法登录用户");%>
void sendHeader(String name, String value)           //设置 HTTP 协议文件头信息
```

例如,下面的语句使 JSP 页面每隔 2 秒自动刷新,常用于制作聊天室。

```
<%response.setHeader("refresh","2");%>
void addHeader(String name, String value)    //添加 HTTP 协议文件头信息,返回给客户端
void addCookie(Cookie name)                  //添加 Cookie 对象,保存给客户端的用户信息
void sendRedirect(String location)           //重新定向 URL
```

5.3.3 输出结果信息的 out 对象

1. 编写使用 out 对象的 JSP 文件

下面的例子可由用户输入一个整数,该页面程序会将这个数的 0～10 次幂计算并显示到页面上。

例 5.5 一个计算用户输入整数幂的 JSP 文件(5-5.jsp 文件)。

```
<%--客户端 html 界面--%>
<center>
<form name=form1 method=get action="5-5.jsp">
显示<input type=text name=n1>的幂值
<input type=submit value="确定"><br>
</form>
<%--提交参数 n1--%>
<%String n1=request.getParameter("n1");%>
<%--服务器端 JSP 获取信息并显示处理结果--%>
<%!int n=2,s=1;%>
<%if (n1!=null)
  {n=Integer.parseInt(n1);
   s=1;
  }%>
<font color="red"><%=n%>的幂值</font>
<table border=2>
<%--绘制表格--%>
  <tr><td>幂次</td><%--表格标题--%><td><%=n%>的幂值</td></tr>
      <%for (int i=0; i<=10; i++) {%>
        <tr><%--表格一行--%>
        <%--第 1 列幂值数--%><td><%=i%></td>
        <%--第 2 列计算结果--%><td><%out.print(s); s=s*n;%></td>
        </tr>
      <% } %>
</table>
</center>
```

在浏览器中输入 http://localhost:8080/myjsp/5-5.jsp,在文本框输入 10,单击“确定”按钮,可看到运行结果如图 5.26 所示。

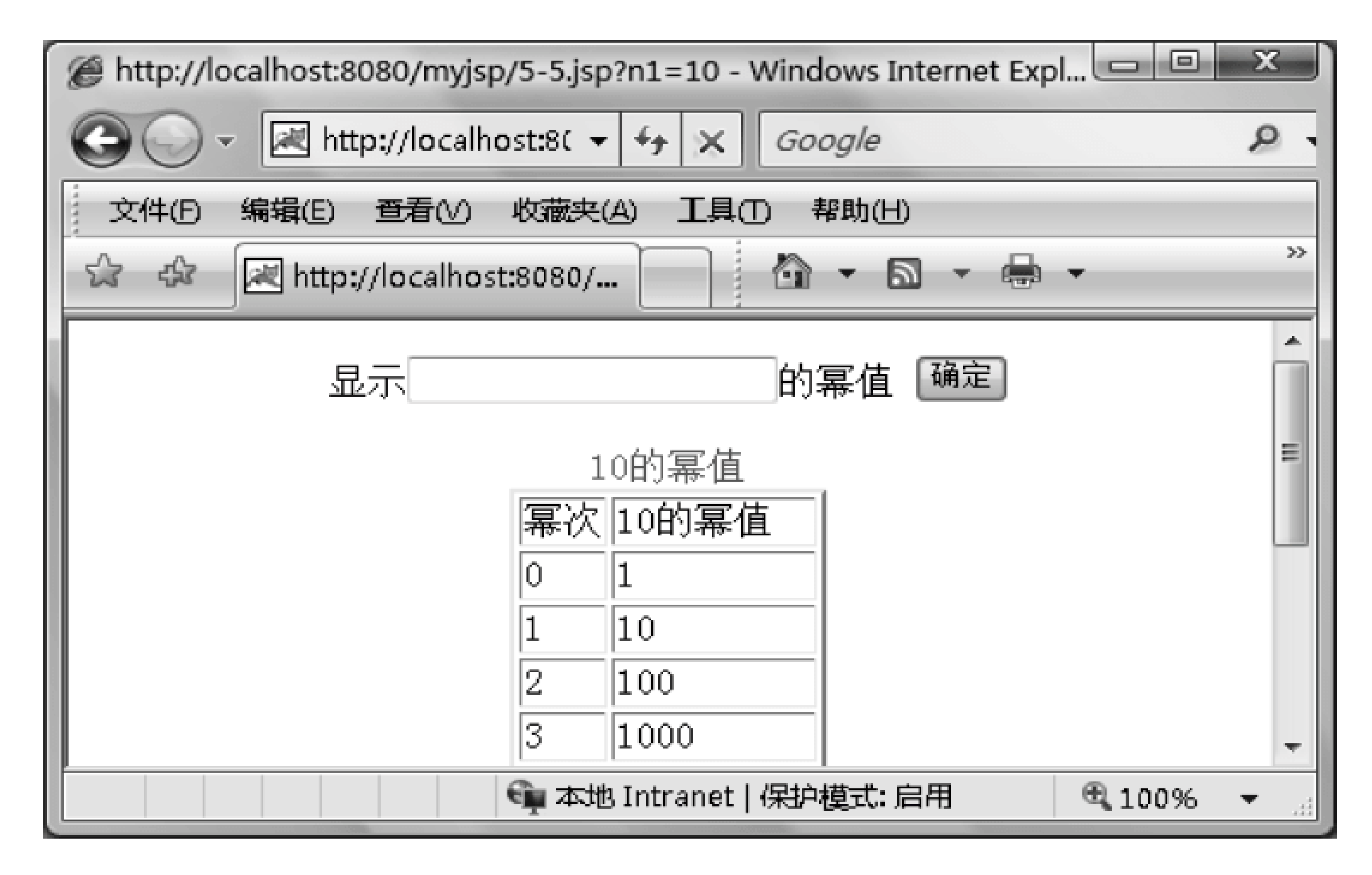


图 5.26 5-5.jsp 运行结果

2. 知识点归纳

1）out 对象

out 对象主要用于向客户端输出数据。out 对象是 java. servlet. JspWriter 类的子类对象。out 对象主要应用于 java 代码片段中，它通过 JSP 容器自动转换为 java. io. PrintWriter 对象，将 JSP 文档的输出信息发送到客户端的浏览器上。

2）out 对象常用方法

```
void print(String s)         //输出字符串信息的方法
void printIn(String s)       //换行输出字符串信息的方法
```

使用 JSP 表达式可以直接输出表达式的值，out 对象可输出 java 代码片段运行的结果，out 对象比较灵活。

3）数据类型的转换

将字符型数据转换为整数型数据可使用 Integer. parseInt(String s)方法。

4）JSP 代码和 HTML 代码混合使用的方式

从本例可以看出，该 JSP 文件里存在 JSP 代码和 HTML 代码混合使用的情况，可在 HTML 代码设计的表格标记中包含 JSP 代码，达到使用 HTML 代码设计表格显示方式、使用 JSP 代码执行计算结果。二者相结合即可在表格中显示 JSP 处理后的计算结果。

5.3.4 保存用户信息的 session 对象

session 对象是在第一次运行 JSP 页面时由 HttpSession 类自动创建的。session 对象用来保存用户信息，以便跟踪用户的操作状态，对通过多个页面完成的 Web 应用非常有用。

为说明 session 对象的具体应用，下面使用三个页面文件组成的 Web 应用。

1. 输入用户姓名的 HTML 文件

例 5.6 创建一个用来输入用户姓名的静态页面文件(5-6.htm 文件)。

```
<html>
<head>
<title>输入信息页面</title>
</head>
<form method=post action="5-6.jsp">
输入姓名<input type="text" name="yhm">
<input type="submit" value="提交" >
</form>
</html>
```

2. 创建保存用户信息的 JSP 文件

例 5.7 创建 5-6.jsp 文件,其功能之一是通过 request 对象提取 5-6.htm 表单中的 yhm 信息,将它保存在 name 变量中,然后将这个 name 值保存到 session 对象中。功能之二是输入"你喜欢吃什么?"问题的答案。

```
<%@page language="java" %>
<%!String name=""; %>
<%
name=request.getParameter("yhm");
session.setAttribute("yhm", name);
%>
你的姓名是:<%=name%><p>
<form method=post action="5-7.jsp">
  你喜欢吃什么?
<input type="text" name="food"><P>
<input type="submit" value="提交">
</form>
```

3. 创建显示各种信息的 JSP 文件

例 5.8 创建 5-7.jsp 文件,它的主要任务是显示问答结果。先通过 session 对象提取 yhm 的信息并显示它,以此说明虽然该信息在第一个页面输入,但通过 session 对象得以保留。5-7.jsp 的另外一个任务是提取在 5-6.jsp 页面中用户根据问题输入的答案。

```
<%@page language="java" %>
<%!String name=""; %>
<%@page language="java" %>
<%!String food1="";%>
<%
food1=request.getParameter("food");
```

```
String name1= (String)session.getAttribute("yhm");
%>
你的姓名是：<%=name1%>
<P>你喜欢吃：<%=food1%>
```

在浏览器中输入 http://localhost:8080/myjsp/5-6.htm，运行结果如图 5.27 所示，输入一个姓名，例如 shlp 后单击“提交”按钮，可看到图 5.28 所示页面，如果在文本框输入“葡萄”后单击“提交”按钮，可看到图 5.29 所示页面。

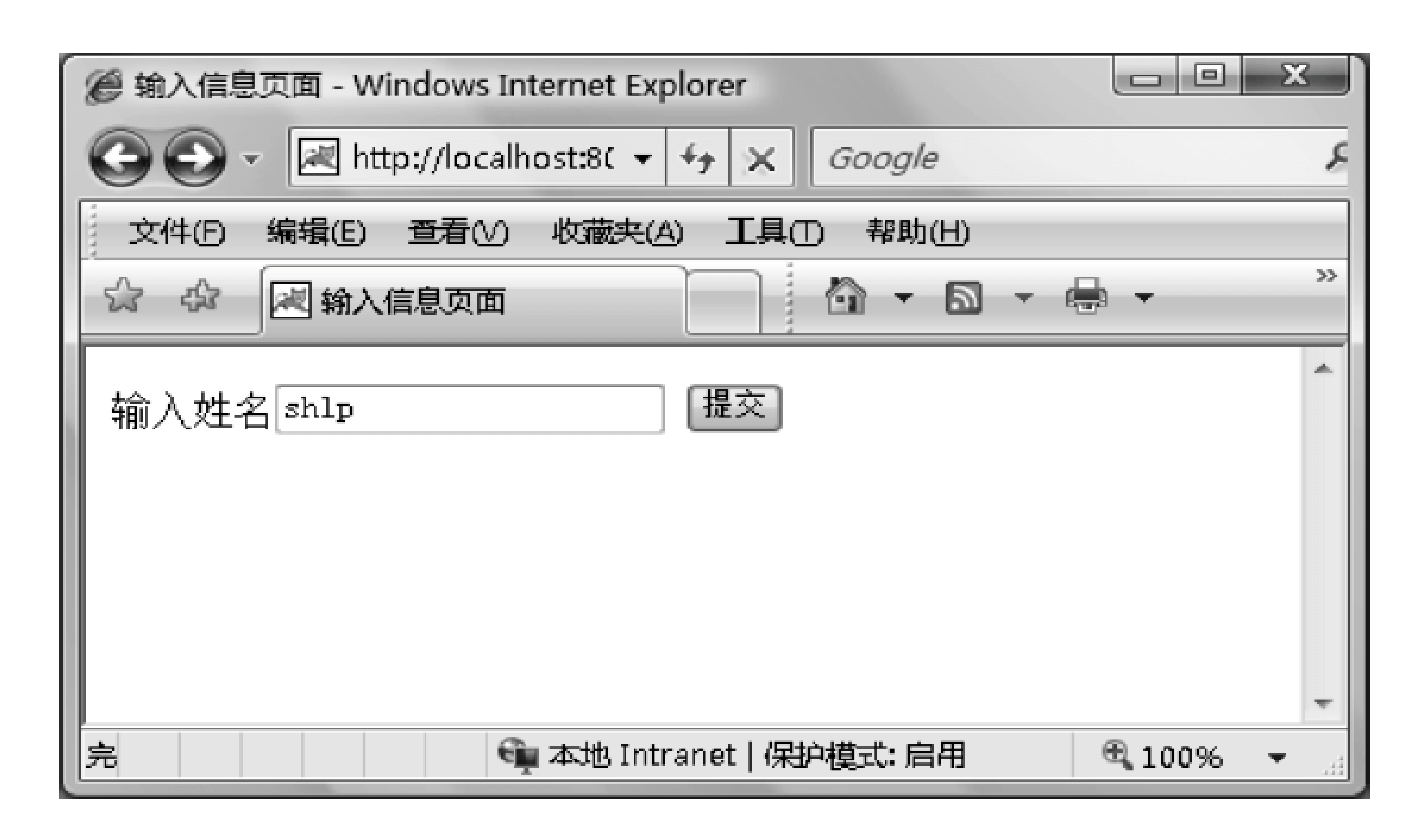


图 5.27　输入姓名

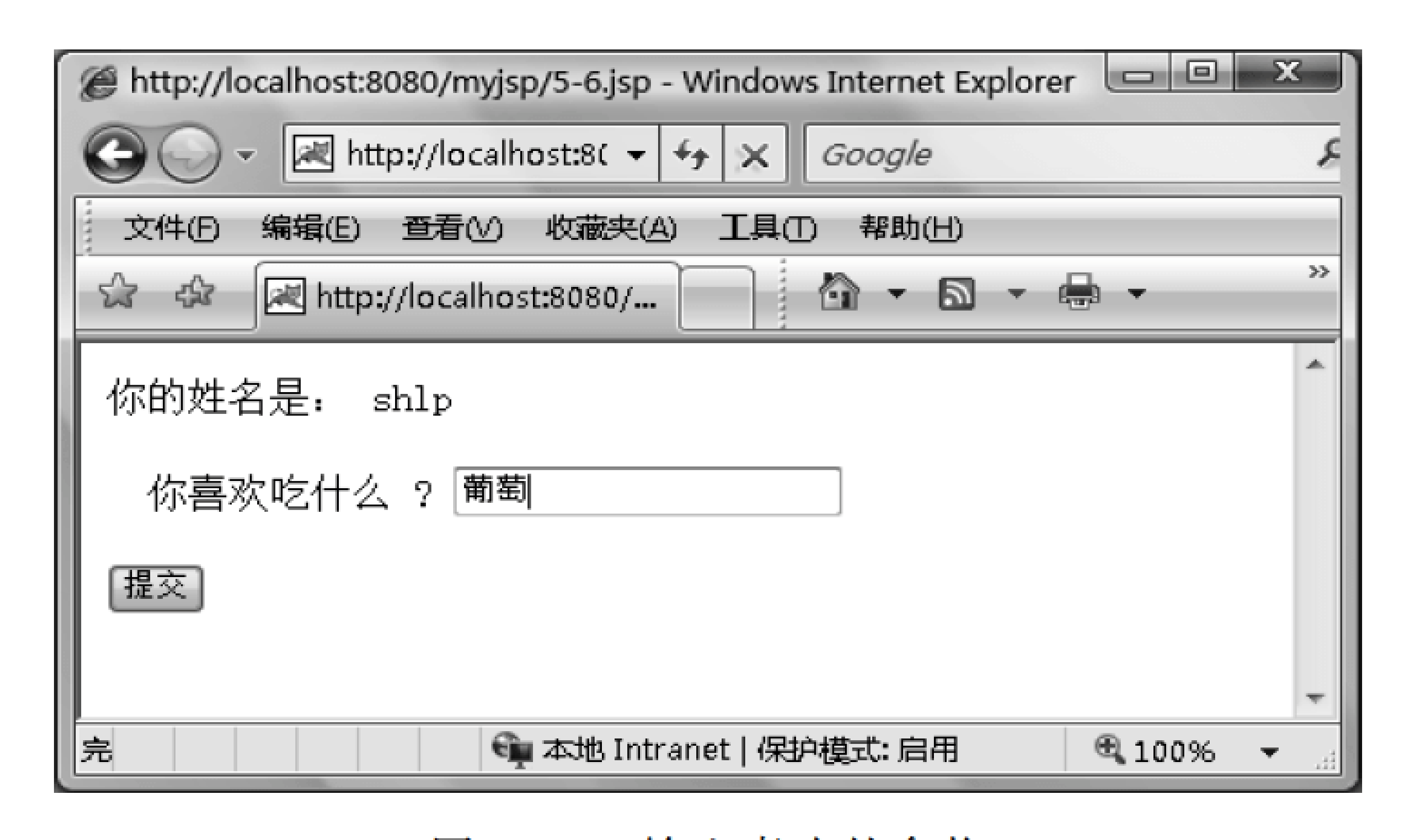


图 5.28　输入喜欢的食物

4. 知识点归纳

1）使用 session 对象保存用户信息的方法

在例 5.8 中通过 setAttribute ("变量名"，变量值)方法定义了 session 类型的变量 yhm，通过 getAttribute ("变量名")方法获取 yhm 变量。session 类型的变量 yhm 用来保存用户输入的用户名变量 name。

2）session 变量的特点

session 变量是由 session 对象专门创建的用于保存客户端信息而分配给用户的变



图 5.29　显示输入的信息

量，是用户的私有变量，每个上网的用户都有但各不相同。当用户第一次在浏览器上访问Web 服务器发布目录下的 JSP 网页时，Web 服务器会自动创建一个 session 对象，并分配唯一的 ID 号。session 类型变量保存在服务器端，ID 号保存在客户端的 Cookie 中。

一个客户端可以有多个不同 session 对象的 ID 号，ID 号和 Web 服务器的 session 对象相对应。由于 session 对象在会话期间是一直有效的，因此 session 对象保存的变量对其他页面也有效。用户可以将需要传递的一些信息保存到 Session 对象中，以便在访问其他页面时使用这些信息。

3）session 对象具有生命周期

session 对象的生命开始于服务器为某个用户建立 session 对象，它的生命结束于服务器所内定的或设置的时间期限。例如，Tomcat 系统默认时间为 30 分钟，如果用户在生成 session 之后，超过这个时间而没有向服务器发出进一步的请求，就会自动被服务器从系统中清除。如果页面需要登录才能使用，用户需要重新登录才能向服务器发出请求。

4）session 对象的常用方法

- boolean isNew() ——判断 session 对象是否是新创建的对象。
- Void setMaxInactiveInterval(int k) ——设置 session 的生命周期。k 的单位为秒。如果时间超过该时间，将不能再对该 session 进行任何操作。
- void putValue/setAttribute(String name，Object value) ——定义名称为 name，值为 value 的 session 变量。
- long getCreationTime()——返回 session 的创建时间。
- long getLastAccessedTime()——返回当前 session 最后一次被访问的时间。
- String getId()——取得 session 的 ID 号，它是服务器创建该 session 时分配的，是唯一的。
- String getValue/getAttribute (String name) ——取得 session 类型变量的值。如果参数 name 在当前 session 中不存在，则返回 null。
- getAttributeNames() ——用于获取 session 对象中存储的每一个属性对象，结果

返回为一个 Enumeration 类对象。

- removeAttribute(String name)——用于删除名称为 name 的属性对象。
- Void invalidate()——设置当前 session 无效，其后不能访问该对象。

5.3.5 保存公共信息的 application 对象

一个 Web 应用通常由多个 HMTL 文件、JSP 文件、JavaBean、数据库文件等资源共同组成，当 Tomcat 启动时，JSP 容器会自动创建一个 application 对象，保存与 Web 应用有关的 Web 服务器名、JSP 容器等信息。application 对象一旦创建，它将一直存在，直到 Web 服务器关闭。因此，application 对象可以实现 Web 应用中多个用户间的数据共享，好像 Web 应用的公共变量一样。

1. 创建 Web 应用的第一个页面

例 5.9 利用 application 对象获得服务器端信息的 JSP 文件(5-8.jsp)。

```
<%
    out.println("JSP 容器:"+application.getServerInfo()+"<br>");
  String filename=request.getServletPath();    //获得本文档名
    out.println("JSP 文档:"+application.getRealPath(filename)+"<br>");

String s=(String)application.getAttribute("count");
    if (s==null)                                    //设置 application 的 count 属性值为 1
        {application.setAttribute("count","1");
        out.println("欢迎你!你是本网站第 "+1+" 位访问者。<br><br>");
}
    else
        {s=(Integer.parseInt(s)+1)+"";
            application.setAttribute("count",s);
            out.println("欢迎你!你是本网站第 "+s+" 位访问者。<br><br>");
}
    String id=session.getId();
    out.println("session 的 ID 号:"+id+" "+"<br>");
    if (session.isNew())
        {out.println("该 session 是新创建的。"+"<br>");}
    int k=36000;
    session.setMaxInactiveInterval(k);
    long n=session.getCreationTime();
    out.println("创建时间:"+(new java.util.Date(n)).toString()+"<br>");
    long m=session.getLastAccessedTime();
    out.println("访问时间:"+(new java.util.Date(m)).toString()+"<br>");
    session.setAttribute("number","1");             //设置 session 的 number 属性值为 1
    out.println("这是第 1 页。<br>");
```

```
%>
<a href="5-9.jsp">下一页</a>
```

2. 创建 Web 应用的第二个页面

例 5.10 利用 application 对象获得服务器端信息的 JSP 文件(5-9.jsp):

```
<%
  out.println("JSP容器:"+application.getServerInfo()+"<br>");
  String filename=request.getServletPath();          //获得本文档名
  out.println("JSP文档:"+application.getRealPath(filename)+"<br>");
  String s=(String)application.getAttribute("count");
  out.println("你好!你是第 "+s+" 位访问者。<br><br>");

  String id=session.getId();
  out.println("session的ID号: "+id+" "+"<br>");
  String nb=(String)session.getAttribute("number");
  if (nb==null)
    {session.setAttribute("number","1");}
  else
    {nb=(Integer.parseInt(nb)+1)+"";
      session.setAttribute("number",nb);}
  out.println("这是第 "+nb+" 页。<br>");
  session.invalidate();                              //设置当前 session 无效
%>
```

在浏览器中输入 http://localhost:8080/myjsp/5-8.jsp,运行结果如图 5.30 所示,单击"下一页"超链接,可看到如图 5.31 所示页面,根据浏览次数不同,其中的数字会改变。这里其实创建了一个计数器,用来显示用户浏览页面的次数。

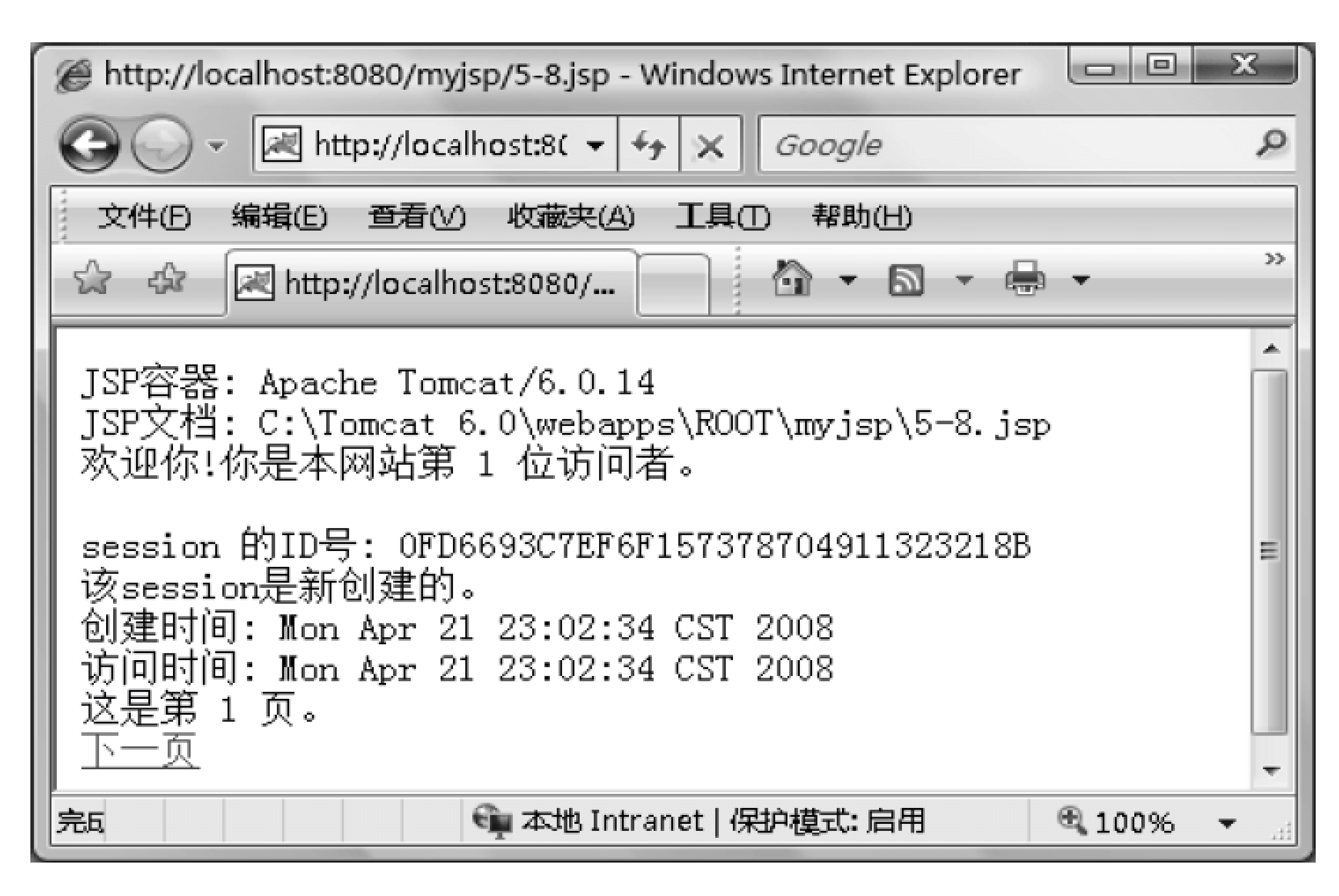


图 5.30 Web 应用的第 1 页

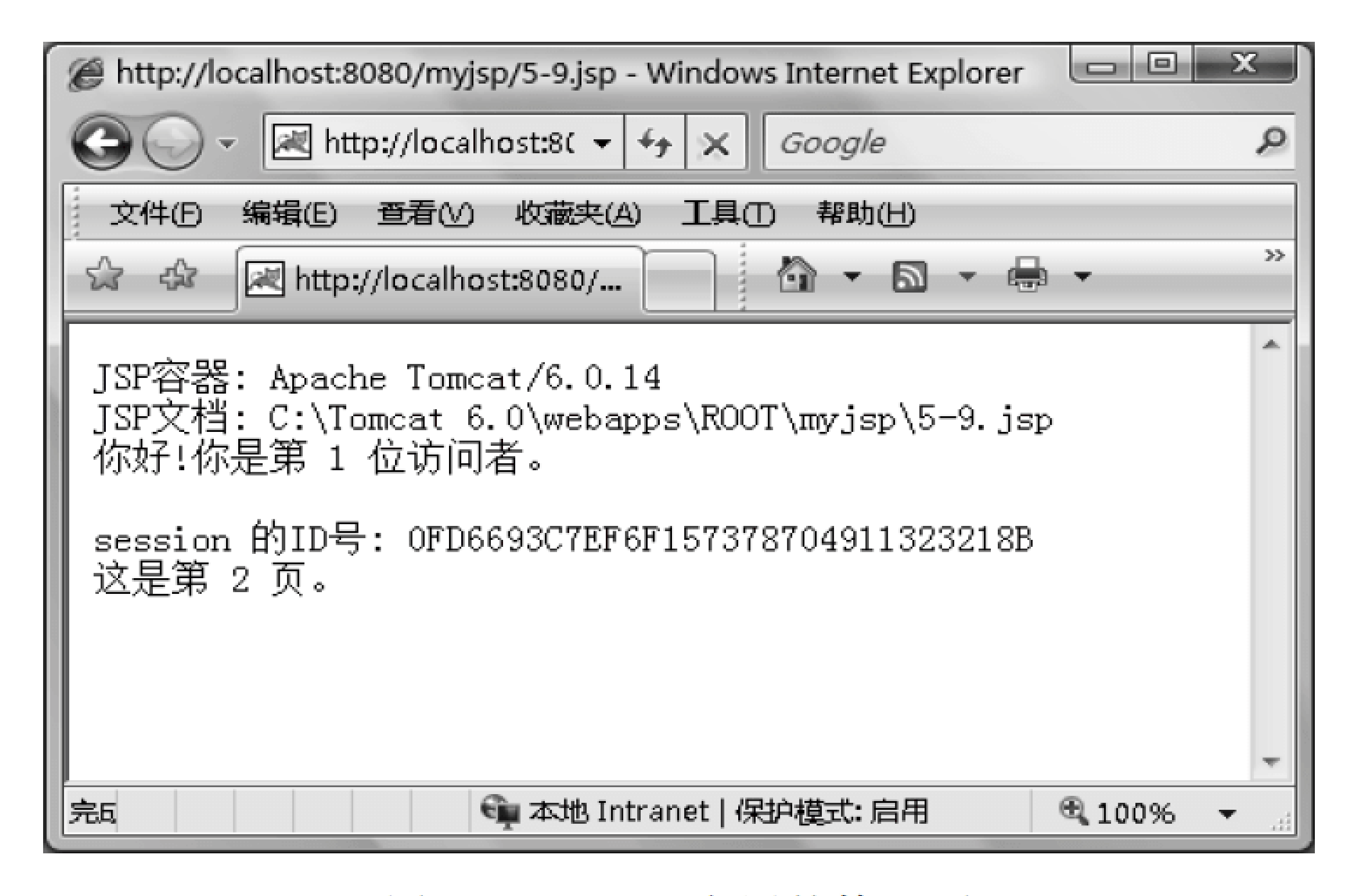


图 5.31　Web 应用的第 2 页

3. 知识点归纳

1）application 对象和 Session 对象的区别

application 对象保存的是不同用户之间共用的数据变量，所以不同用户都可以读取到 application 对象的变量值。而 Session 对象存储的是不同用户的私有数据，服务器端为每个 session 都要附加上 ID 识别码，所以 Session 是私有的，只属于个别用户。

从例 5.9 与 5.10 可以看到使用 application 对象设置 count 属性保存用户第几次访问该网页，使用 Session 对象设置 number 属性值保存当前访问的是 Web 应用的第几页。

2）application 的生命期

Application 对象从 Web 服务器启动开始生成，到 Web 服务器关闭消失。

3）application 对象的常用方法

- String getRealPath ()——返回 filename 文件所在的绝对路径。
- void setAttribute (String name,String value) ——设置名称为 name 值为 value 的 application 类型变量。
- String getServletInfo()——返回当前使用的 Web 服务器(Servlet 编译器)及版本号信息。
- Object getAttribute (String name) ——取得 application 类型变量的值。
- getAttributeNames() ——用于获取 application 对象中存储的每一个属性对象，结果返回为一个 Enumeration 类对象。

其他内部对象还有 Exception、PageContext、Config、Page 等。

5.4　JSP 的动作标记

在 JSP 文件中，还可以包含 JSP 动作标记。使用 JSP 动作标记可以动态地插入其他网页文件、重用 JavaBean 组件、把用户重定向到另外的页面、为 Java 插件生成 HTML 代码等。

本节的内容主要介绍 JSP 动作标记 jsp:include、jsp:forward、jsp:plugin、jsp:useBean 等的使用方式。

5.4.1 jsp:include 动作标记

1. 语法格式

jsp:include 动作标记可以把指定文件插入到正在生成的 JSP 页面文件中。

其语法格式如下：

```
<jsp:include page="relative URL" flush="true"/>
  <jsp:param name="属性名" value="属性值"/>
</jsp:include>
```

例如：

```
<jsp:include page="news/new1.html" flush="true"/>
<jsp:include page="news/new2.html" flush="true"/>
<jsp:include page="news/new3.jsp" flush="true"/>
```

其中，page 属性说明文件的相对 URI 地址，flush 是否立即刷新缓存。引入文件时可通过<jsp:param>子标记同时引入属性变量。

2. 路径的写法

在 JSP 中将使用相对 RUL 规范，即相对路径。它分两种类型：

1）相对路径

不以斜杠"/"开头的路径，例如"news/new1.html"。"../new.jsp"表示 new.jsp 存放在相对当前页面的上一层路径，"news/new2.html"表示 new2.html 存放在相对当前页面的下一层路径 news 中。

2）绝对路径

以斜杠"/"开头的路径，例如"/WZKF/myjsp/news/new1.html"。new1.html 相对它的 ServletContext 对象提供的根路径。它以某个 Web 服务 RUL 路径(虚拟目录的根目录，下节介绍)开始。

注意：文件系统目录中的斜杠为"\"。例如"C:Tomecat 6.0"。

3. jsp:include 动作与 include 指令

1）引入时间

include 指令语句，在 JSP 文件被转换成 Servlet 类文件(.class 文件)时引入被包含文件的，仅在 JSP 文件更新编译时才会再次更新被包含的文件。jsp:include 动作在 JSP 文件被请求时引入被包含文件。

2）引入文件类型

include 指令语句可以引入 HTML、JSP 等类型的文件。include 指令在编译时将引

入的文件一起解释执行。jsp:include 动作只处理后缀名为.jsp 的文件,其他类型文件原样输出交浏览器解释。

3) 特点

jsp:include 动作标记引入文件的时间决定了它的运行效率要稍微差一点,而且被引入的文件不能包含某些 JSP 代码(例如不能设置 HTTP 头),但动态性好,能在每次 JSP 文件被请求时更新其内容。所以,如果引入文件的内容经常要变可使 jsp:include 动作。文件内容基本不变可用 include 指令来引入。

5.4.2 jsp:forward 动作标记

jsp:forward 动作标记可从一个 JSP 文件向另一个文件传递一个包含用户请求的 request 对象。jsp:forward 标记只有一个属性 page。page 属性说明相对的 URL。page 的值既可以直接给出,也可以在请求的时候动态计算,其语法格式如下所示。

```
<jsp:forward page="relative URL" flush="true"/>
  <jsp:param name="属性名" value="属性值"/>
</jsp:forward>
```

例如:

```
<jsp:forward page="/myjsp/5-1.jsp"/>或<jsp:forward page="<%=表达式%>"/>
<jsp:param name="username" value="lp"/>
</jsp:forward>
```

通过<jsp:param>子标记向目标文件(5-1.jsp)传送了参数和值,参数名为 username,值为 lp。

5.4.3 jsp:plugin 动作标记

jsp:plugin 动作标记用来根据浏览器的类型,插入 Java Applet 与 JavaBean。

例如,下面的语句可在 ISP 页面上插入一个 Java Applet 类 Hello.class。

```
<jsp:plugin type="applet" code="Hello.class" height="40" width="320" >
  <jsp:params>                                   //指定 applet 的参数
  <jsp:param name=bgcolor value="ffffcc"/>
  <jsp:param name=fgcolor value="ff0000"/>
  </jsp:params>
  <!--如果无法载入 Applet 时显示的信息-->
  <jsp:fallback>无法加载 Applet</jsp:fallback>
</jsp:plugin>
```

当 JSP 文件被编译后,送往浏览器时,<jsp:plugin>标记将会根据浏览器的版本替换成<object>或者<embed>元素。

5.4.4 jsp:JavaBean 动作标记

1. 什么是 JavaBean

JavaBean 是基于 Java 的组件技术，它提供了创建和使用以组件形式出现的 Java 类的方法。编写 JavaBean 的目的是为了重复使用。JSP 对于在 Web 用中集成的 JavaBean 组件提供了完善的支持。这种支持不仅能缩短开发时间(可以直接利用经测试和可信任的已有组件，避免重复开发)，也为 Web 应用带来了更多的功能，例如，JavaBean 可以帮助 JSP 执行复杂的计算任务、负责与数据库进行交互以及提取数据等。

2. 创建 JavaBean 的步骤

1) 编写 JavaBean 源程序(TaxRate.java 文件)

例 5.11 使用 Java 语言创建一个名称 TaxRate 的 JavaBean。它具有两个变量：Product(产品)和 Rate(税率)。两个 set 方法分别用来设置这两个变量，两个 get 方法用于提取这两个变量的值。在实际应用中，这种 JavaBean 一般应当从数据库提取税率值，此处简化了这个过程，允许任意设定税率。

使用 Java 语言编写的 JavaBean(TaxRate.java 文件)代码如下：

```
package tax;
public class TaxRate {                                    //创建 TaxRate 类
String Product;                                           //声明 Product 变量
double Rate;                                              //声明 Rate 变量
public TaxRate() {                                        //不带参数的构造方法
this.Product="A001";
this.Rate=5;
}
public void setProduct (String ProductName) {             //设置 Product 值的方法
this.Product=ProductName;
}
public String getProduct() {                              //获取 Product 值的方法
return (this.Product);
}
public void setRate (double rateValue) {                  //设置 Rate 值的方法
this.Rate=rateValue;
}
public double getRate () {                                //获取 Rate 值的方法
return (this.Rate);
}
}
```

这是一个 Java 源程序，它定义了一个 tax 包，在包中定义了 TaxRate 类及其变量和方法，它没有 main()方法。这种 Java 类称为 JavaBean。

一个标准的 JavaBean 具有以下三个特点：

- JavaBean 是一个公共(public)类；
- JavaBean 有一个不带参数的构造方法；
- JavaBean 具有 setXXX()方法可以设置变量的值，同时具有 getXXX()方法可以获取变量的值。

2）编译生成字节码文件的 JavaBean(TaxRate.class)

将 TaxRate.java 源程序编译为 TaxRate.class 字节码文件。如果没有其他编译环境，可参考以下编译方法：

(1）首先确定 Windows 系统中的环境变量 Path 与 classpath 为 C:\sdk\jdk\bin;(根据 JDK 保存的路径进行设置)，以保证可以对 Java 源程序进行编译。

(2）在命令提示符窗口或 MS-DOS 窗口进入 TaxRate.java 程序所在目录。

(3）输入编译器文件名和要编译的源程序文件名“javac TaxRate.java”，按 Enter 键即开始编译(注意：文件名中大小写不能错，否则运行会出问题)。

如果源程序没有错误，则屏幕上没有输出，可输入 dir 按 Enter 键后可在目录中看到生成了一个同名字的.class 文件 TaxRate.class，否则将显示出错信息。

3）保存 JavaBean(TaxRate.class)

将编译后的 TaxRate.class 文件放在 C:\Tomcat 6.0\webapps\ROOT\WEB-INF\classes\tax 目录下，WEB-INF\classes 是专门存放 JavaBean 的子目录，tax 是存放 TaxRate.class 的子目录。

3. 在 JSP 中使用 JavaBean

例 5.12 编写使用名称为 TaxRate 的 JavaBean 的 JSP 文件(TaxRate.jsp)，代码如下：

```
<html>
<%@page language="java"%>
<h4>应用 JavaBean 的 JSP 页面</h4>
<jsp:useBean id="taxbean" scope="application" class="tax.TaxRate" />
初始设置的产品号与税率<br>
产品：<%=taxbean.getProduct()%><br>
税率：<%=taxbean.getRate()%>
<p>重新设置产品与税率值<br>
产品：A002<%taxbean.setProduct("A002");%><br>
税率：17<%taxbean.setRate(17);%><p>

显示新设置的产品与税率的值<br>
产品：<%=taxbean.getProduct()%><br>
税率：<%=taxbean.getRate()%>
</html>
```

在浏览器中输入 http://localhost:8080/myjsp/TaxRate.jsp，运行结果如图 5.32 所示。

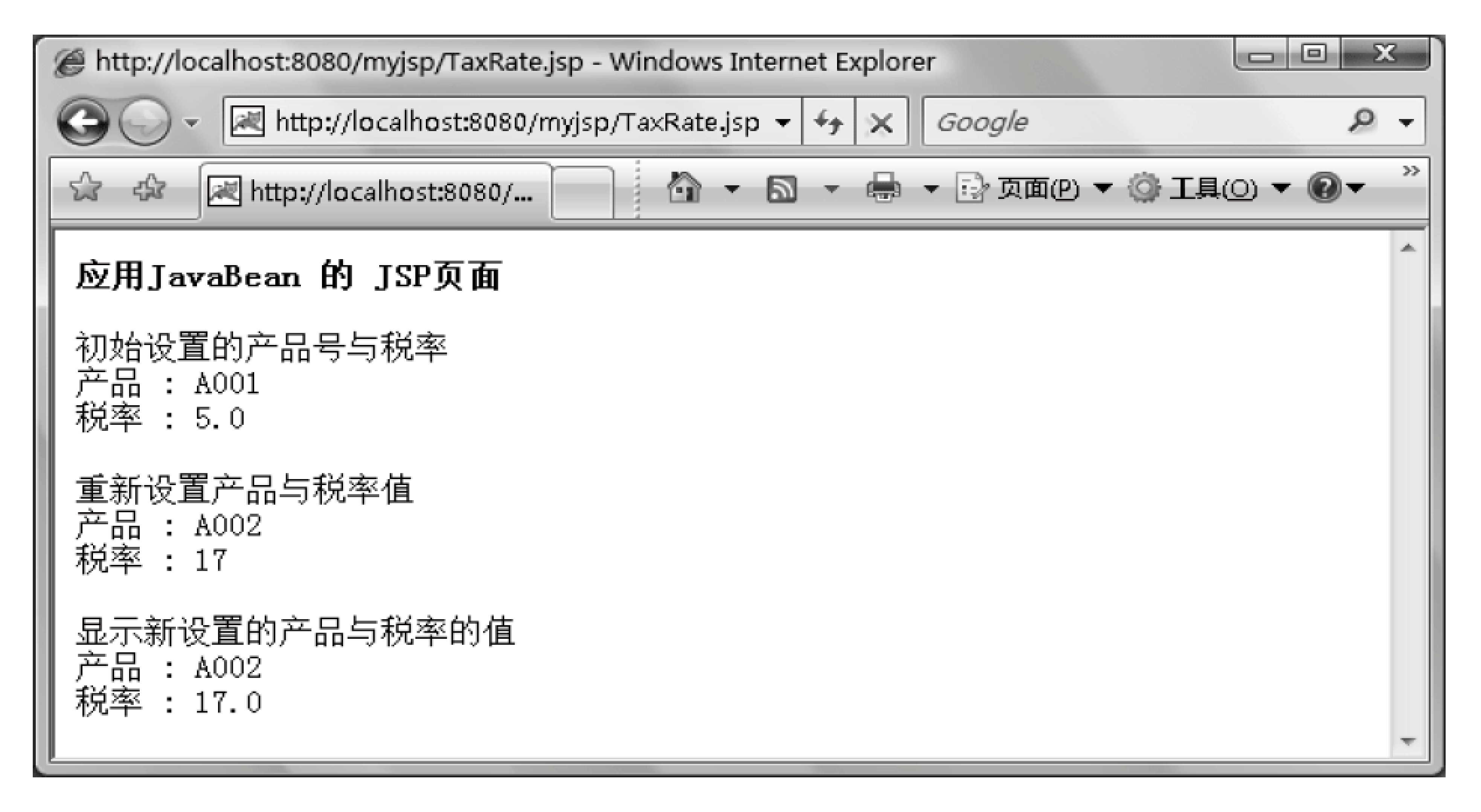


图 5.32 TaxRate.jsp 运行结果

可以看到在 TaxRate.jsp 页面中通过 TaxRate 的 set 和 get 方法重新设置并显示了变量的值。该例没有什么实际意义，只是让读者了解如何创建和使用 JavaBean。

4. 知识点归纳

1）声明并创建 JavaBean 的对象

通过<jsp:useBean>动作标记可以在 JSP 页面中声明并创建一个 JavaBean 的对象。这个功能非常有用，因为它既可以继承 Java 组件重用的优势，又可以发挥 JSP 的方便性。

jsp:useBean 动作标记的语法格式为：

```
<jsp:useBean id="name" class="package.class" scope="scope"/>
```

它表示由<jsp:useBean>创建了一个由 class 属性指定的类的实例对象，并由 id 属性给出实例对象名，由 scope 属性给出 JavaBean 的使用范围。例如下面的语句：

```
<jsp:useBean id="taxbean" class="tax.TaxRate" scope="application" />
```

定义了一个 tax 包中的 TaxRate 类的 JavaBean 对象，其名为"taxbean "。

范围属性 scope 确定 JavaBean 的使用范围，有 4 种参数。application 表示 JavaBean 的使用范围为整个应用程序。page 表示 JavaBean 的使用范围是当前页面，scope 的默认值就是 page，表示该 JavaBean 只在当前页面内可用。request 表示 JavaBean 的使用范围为一个被请求的网页。session 表示 JavaBean 的使用范围为 session 变量内。

2）调用 JavaBean 的变量与方法

在 JSP 文件中通过 JavaBean 的对象名可以直接调用 JavaBean 对象的变量与方法。

例如，通过 taxbean 可以使用 TaxRate 对象的方法 getProduct()得到产品的编号，参见下面的语句：

```
<%=taxbean.getProduct()%>
```

3）更改 JavaBean 变量的值

在 JSP 文件中可以直接更改 JavaBean 中变量的值，例如可使用 TaxRate 对象的 setProduct(参数)方法来更改 TaxRate 对象的 Product 变量的值，参见下面的语句：

```
<% taxbean.setProduct("A002");%>
```

4）存放 JavaBean 的位置

为了要让网络服务器可以找到 JavaBean，需要将 JavaBean 的类文件放在特别位置。一般将 JavaBean 的类文件存放在指定的 JavaBean 包(c:\Tomcat 6.0\webapps\ROOT\WEB-INF\classes\包名)目录下。

5.4.5 jsp:setProperty 与 jsp:getProperty 动作标记

<jsp:setProperty>动作标记用来设置实例化的 JavaBean 对象的属性，在<jsp:useBean>标记出现后使用。例如：

```
<jsp:useBean id="taxbean" scope="application" class="tax.TaxRate" >
<jsp:setProperty name="taxbean" property="Product" value="A003">
<jsp:setProperty name="taxbean" property="Rate" value="13">
```

jsp:getProperty 动作标记表示先提取 JavaBean 对象属性的值，然后转换成字符串在 JSP 页面中输出。

例如：

```
<jsp:getProperty name="taxbean" property="Product">
<jsp:getProperty name="taxbean" property="Rate">
```

它们都有两个属性，name 为实例化的 JavaBean 对象的名字；property 为 JavaBean 对象的属性名。

5.5 创建虚拟目录

上面介绍的 JSP 文件都是保存在 Tomcat 根目录 ROOT 下的，浏览器能够正常运行这些 JSP 文件。如果将 JSP 文件保存在其他文件夹中浏览器是否能正常运行它们呢？

本节的内容主要解决如何在虚拟目录下运行保存在其他文件夹中的 JSP 文件的问题。

5.5.1 虚拟目录、根目录与物理目录

1. 根目录(或称主目录)

启动 Tomcat 服务器后，Tomcat 服务器就可以对用户浏览器提交的网页浏览请求做出响应。为了使 Tomcat 服务器实现这种响应，需要将发布的文件保存在站点的根目录或该目录的子文件夹中。

Tomcat 默认站点的路径为“C:\Tomcat 6.0\webapps\ROOT”，该路径称为 Tomcat 的根目录。在浏览器中输入网址：http://localhost:8080/文件夹/文件名，即可访问“C:\Tomcat 6.0\webapps\ROOT\文件夹”中相应的文件，例如，http://localhost:8080/myjsp/5-1.jsp。

2. 物理目录

在计算机或存储器中存放文件的文件管理系统的路径称为物理目录。例如，可设置“C:\wzkf”物理目录，将编写的网页文件都保存在这里。

3. 虚拟目录

虚拟目录是指在计算机中不存在的路径。但 Tomcat 却将其视为根目录。

4. 别名

别名是虚拟根目录对应物理目录的名称，Tomcat 服务器通过别名运行物理目录下的网页文件。别名可与物理文件夹同名，也可不同。为了方便输入网址，别名一般比真实文件夹名短。例如，“C:\wzkf”文件夹名，可用 jsp 作为别名。这样可以增加程序的安全性，使外人不容易发现文件所在的物理目录。

5.5.2 通过 Tomcat 管理器创建虚拟目录

首先创建物理目录 C:/WZKF。

创建虚拟目录可以参考以下操作步骤：

(1) 在 Windows 界面单击“开始”→“所有程序”→Apache Tomcat 6.0→Tomcat Manager 命令，打开如图 5.33 所示窗口。

(2) 输入用户名，这是安装 Tomcat 时确定的，默认名称为 admin，如果输入了口令还需要输入口令，本书根据安装时定义口令为空，单击“确定”按钮，打开如图 5.34 所示窗口。

(3) 在 Deploy 栏下的 Context Path 文本框输入虚拟目录名称/jsp，如图 5.35 所示窗口。

(4) 在 WAR or Directory URL 文本框输入物理目录路径/D:\WZKF，如图 5.35 所示窗口。

(5) 单击 Deploy 按钮，可看到如图 5.36 所示的虚拟目录信息。一个是创建成功的 OK 信息，一个是在 Applications 栏下虚拟目录的显示，在其右边单击 Undeploy 超链接，可以清除所创建的虚拟目录。



图 5.33　Tomcat 管理器登录窗口

(6) 将 5-1.jsp 文件存放在 C:\WZKF\myjsp 路径下，在浏览器地址栏输入 http://localhost:8080//jsp/myjsp/5-1.jsp，按 Enter 键后将看到如图 5.37 所示页面。

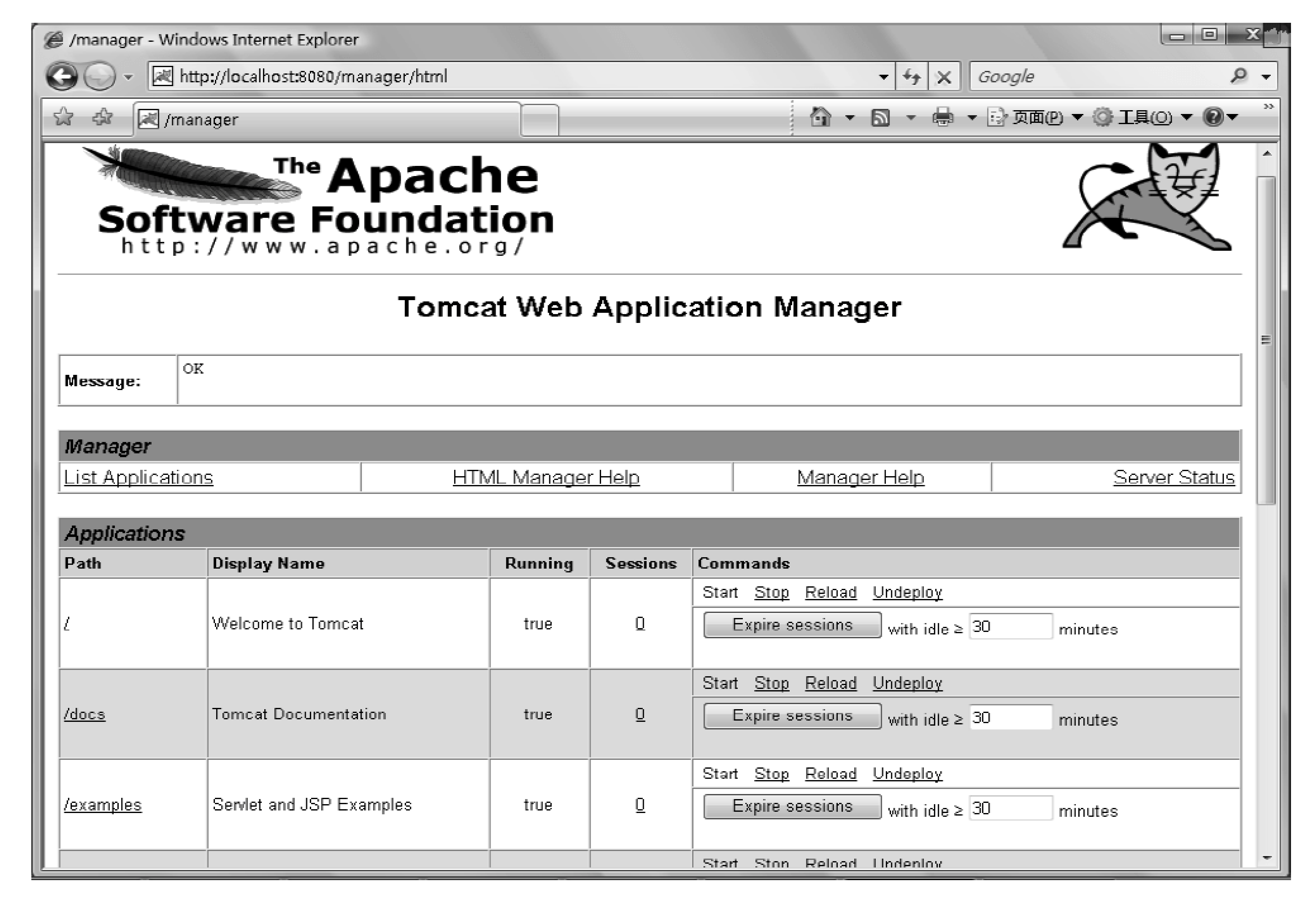


图 5.34　Tomcat Web Application Manager 窗口

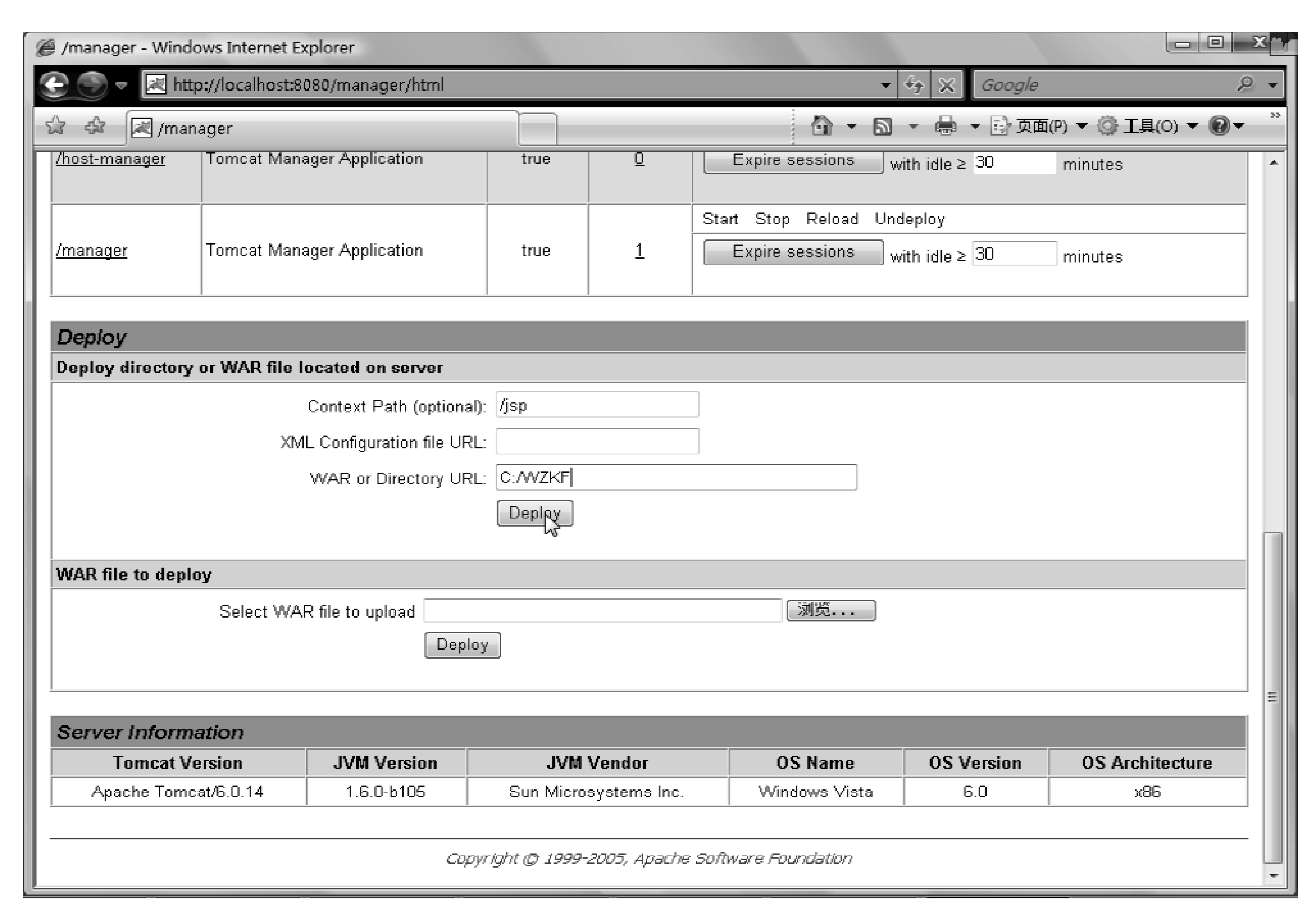


图 5.35　输入虚拟目录名称与物理目录名称

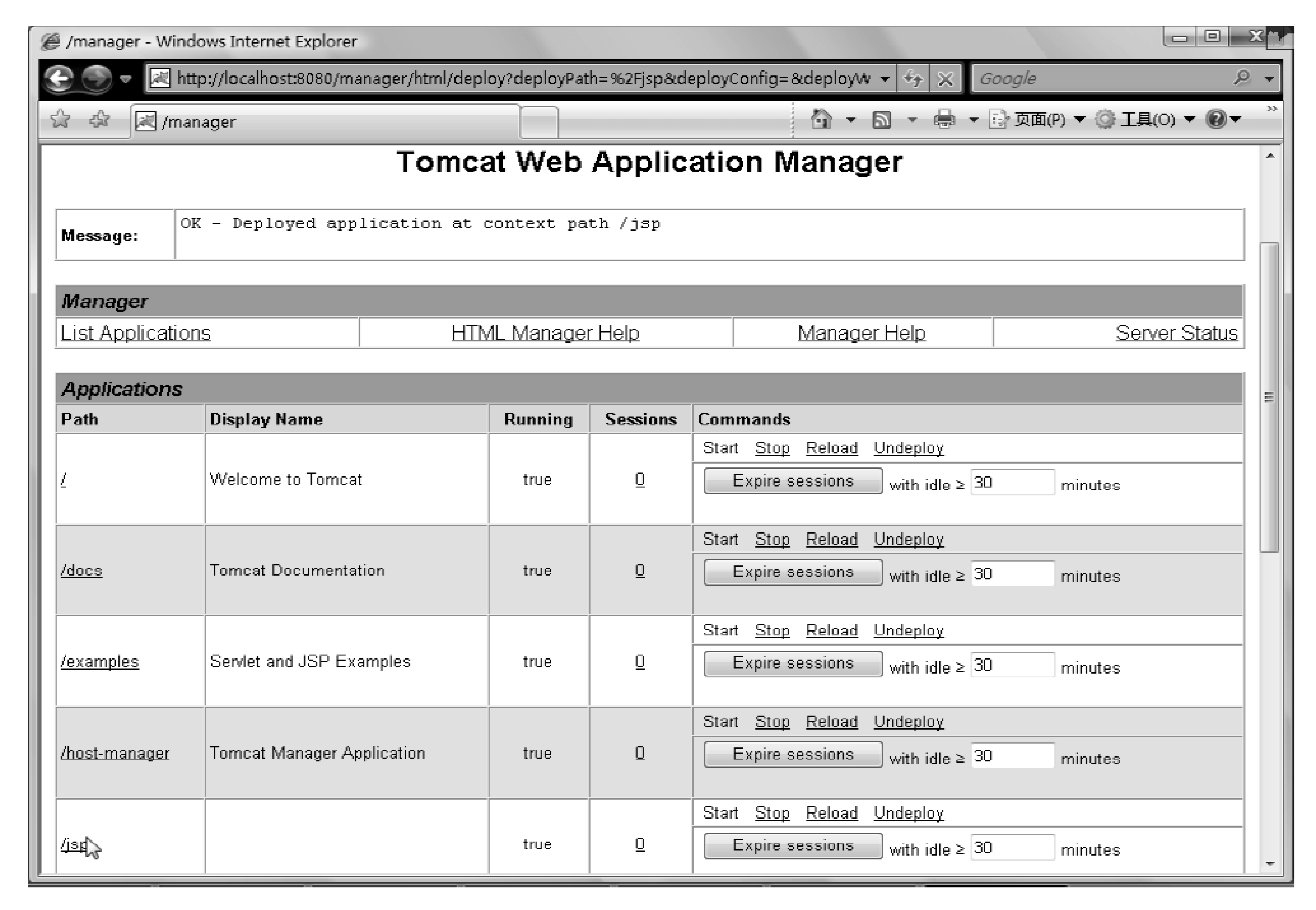


图 5.36 创建虚拟目录成功

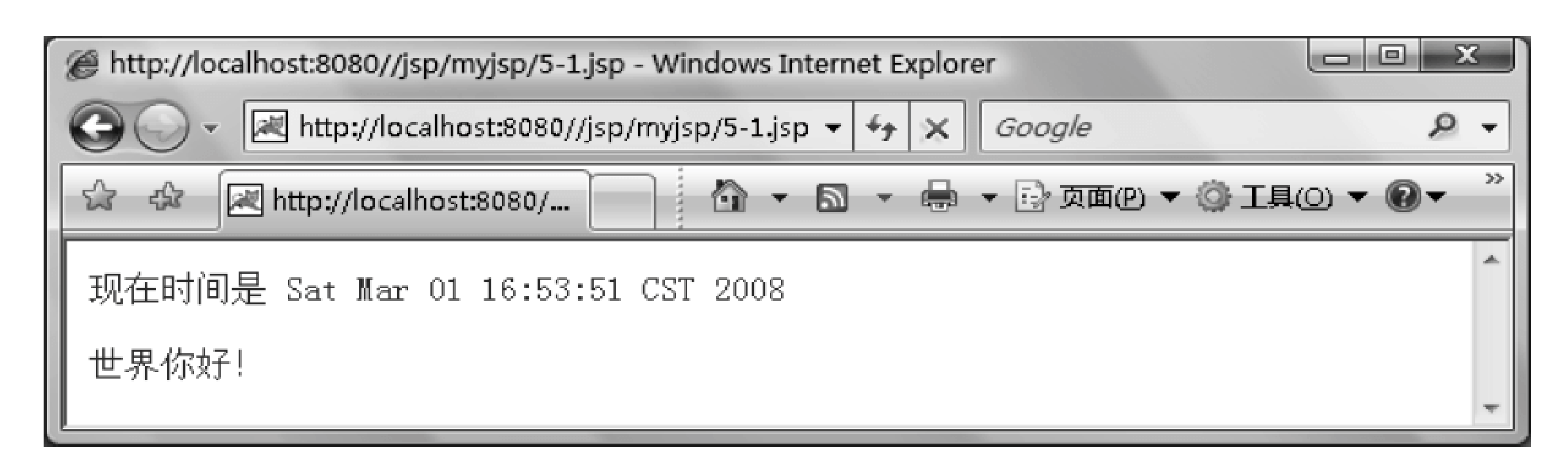


图 5.37 通过虚拟目录运行的 jsp 文件

5.5.3 通过 server.xml 文件创建虚拟目录

通过 server.xml 文件创建虚拟目录可参考如下操作步骤：

(1) 在资源管理器中 Tomcat 6.0 目录下的 conf 文件夹中，找到 server.xml 文件，如图 5.38 所示。

(2) 在网页编辑器中打开 server.xml 文件，在<host></host>标记之间加入如下 Context 标记对，如图 5.39 所示。

```
<Context path="/lp" docBase="c:\lp" debug="0" reloadable="true"></Context>
```

(3) 重启 tomcat 服务器即可通过虚拟目录运行 JSP 文件。

注意：

(1) 通过 server.xml 文件创建的虚拟目录，只能通过 server.xml 文件删除。

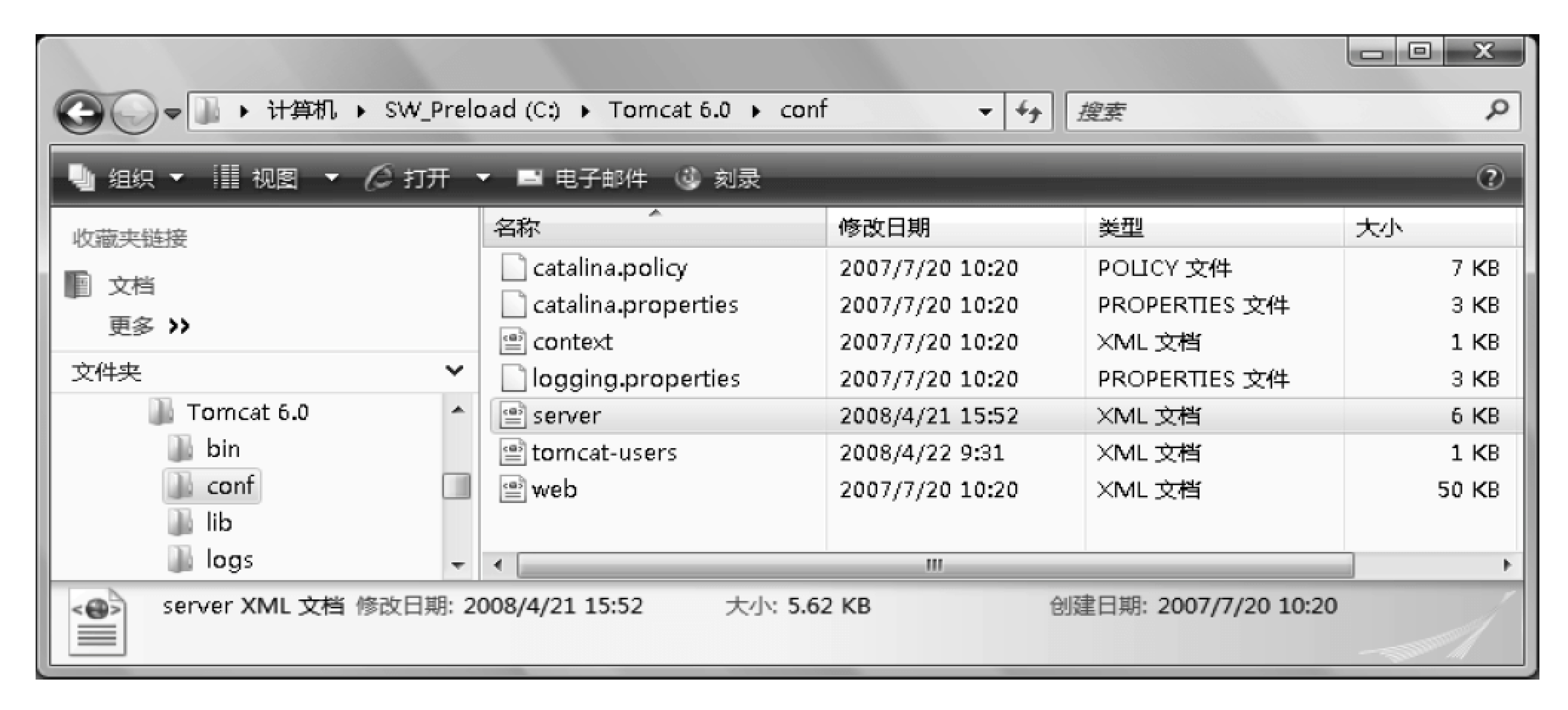


图 5.38 server. xml 文件存放路径

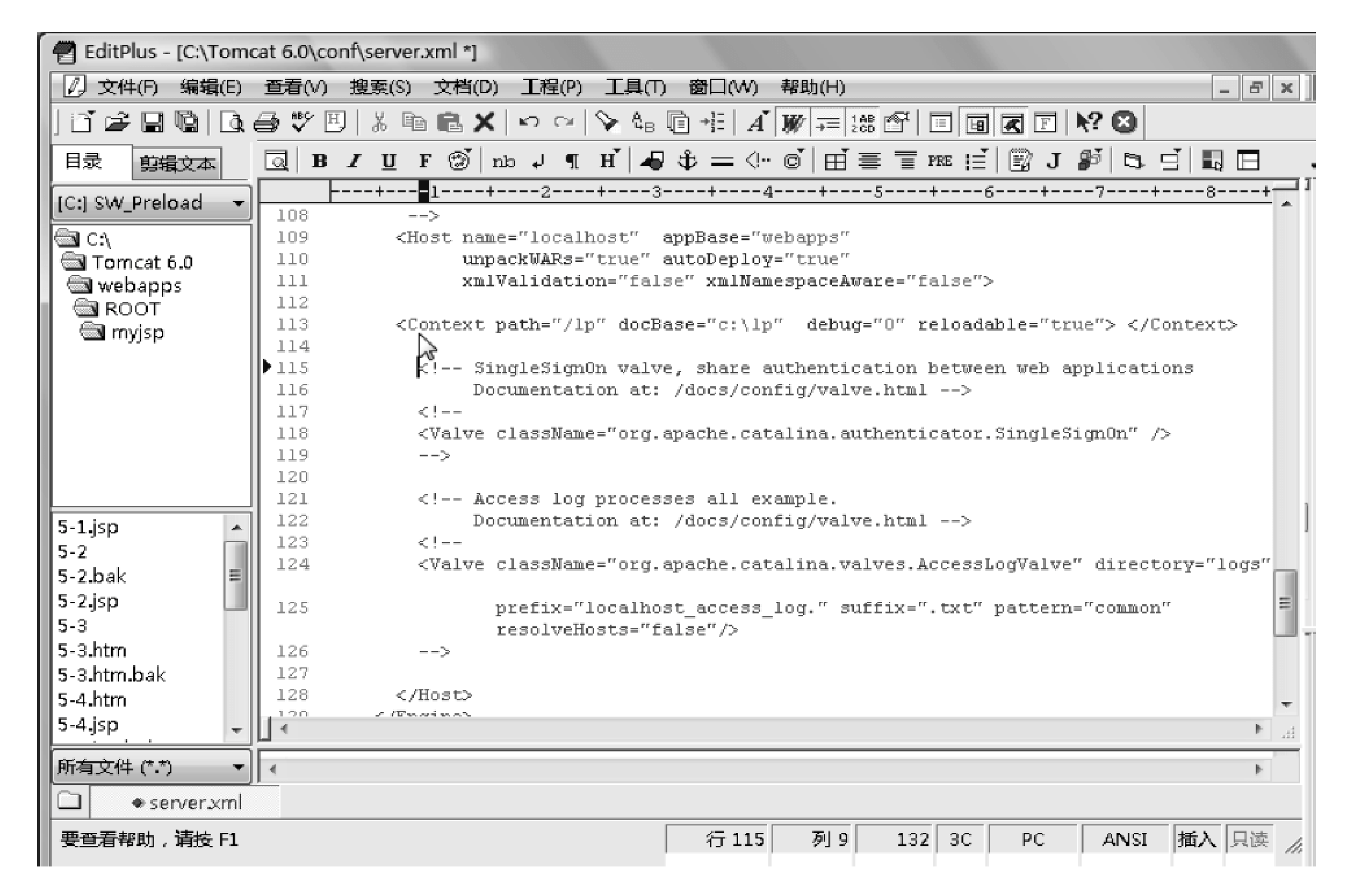


图 5.39 添加 Context 标记对

(2) 如果在连接数据库时使用的是物理地址，需要通过 server. xml 文件来设置虚拟目录。

5.6 思考与练习

5.6.1 思考题

5-1 JSP 由哪些基本元素构成？

5-2 JSP 有哪些常用的内置对象？

5-3 JSP 有哪些动作标记?

5-4 JSP 与 Java 存在什么关系?

5-5 JavaBean 有什么作用?

5.6.2 上机练习

5-1 下载安装 SDK,并设置 Windows 2000/XP 环境中的"环境变量"path 与 classpath,其变量值都为"C:\sdk\jdk\bin;",设置完成后,在 Windows 2000/XP 环境下选择"开始"→"所有程序"→"附件"→"命令提示符"命令,打开 DOS 窗口,在命令提示符下输入 java 或 javac,按 Enter 键后,如果出现其用法参数提示信息,则安装正确。

5-2 编译一个 Java 程序 Hello.java,代码如下:

```
public class Hello {
public static void main(String args[]) {
    System.out.println("欢迎你学习 Java 语言!");
}
}
```

将 Hello.java 保存在"C:\java\程序>"目录下,打开 Windows 的"命令提示符窗口",进入"C:\java\程序>"目录下,输入 javac Hello.java,按 Enter 键。

看当前目录下是否有编译好的 Hello.class 字节码文件。

5-3 下载与安装 Tomcat,设置 JSP 运行环境。创建一个存放 JSP 文件的文件夹,将例 5-1.jsp 复制过去,看是否能在浏览器中显示该 JSP 页面。

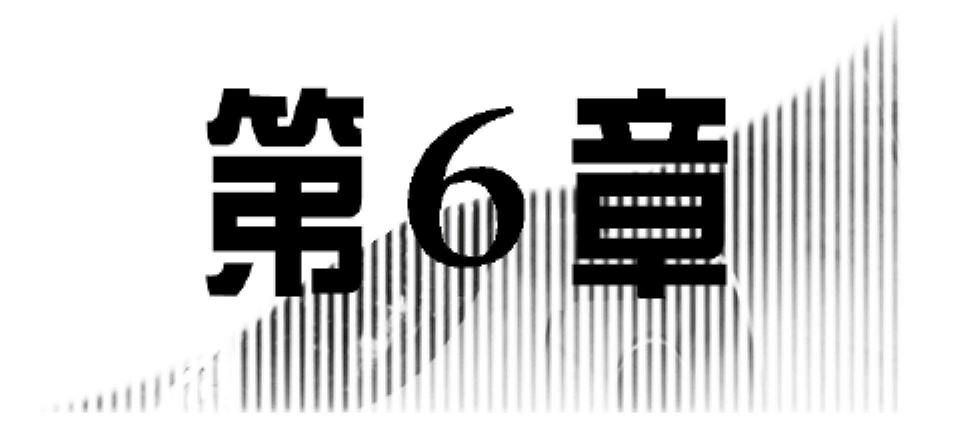
5-4 编写一个包含有本书介绍的 5 种基本语句的 JSP 程序。

5-5 编写一个简单的 JavaBean,并编写一个应用 JavaBean 的 JSP 程序。

5-6 编写一个 HTML 文件,包含一个输入用户名的文本框和输入口令的文本框,编写一个根据输入的用户名和口令显示不同结果的 JSP 页面。当用户名和口令为 test 和 1234 时,显示"欢迎你进入 JSP 学习乐园!",否则,显示"你输入的用户名和口令不正确。"(可参考 JavaScript 中的选择结构语句)。

5-7 编写一个可以输入不同整数 n 的文本框,在当前 JSP 页上显示 $n!$ 计算结果的 JSP 页面。

5-8 编写一个显示 application 和 Session 变量保存的信息的页面。



第6章 使用 JSP 访问数据库

现在每一个人的生活几乎都离不开数据库，如果没有数据库，很多事情都会变得非常棘手，也许根本无法做得到。银行、大学和图书馆就是几个严重依赖数据库系统的地方。在互联网上，使用搜索引擎、在线购物甚至是访问网站地址（http://www...）都离不开数据库。数据库通常都安装在称为数据库服务器的计算机上。

本章主要解决以下问题：

- 什么是 SQL；
- 什么是 Access 数据库；
- 如何创建 Access 数据库并在其中保存数据；
- 什么是 ODBC 与数据源；
- 什么是 JDBC；
- 如何通过 JSP 页面访问数据库。

6.1 结构化查询语言 SQL

什么是 SQL 呢？

SQL 是“Structured Query Language”结构化查询语言的缩写，SQL 是用于对存放在计算机数据库中的数据进行组织、管理和检索的一种工具；SQL 是针对一种特定类型的数据库——关系数据库使用的。控制关系数据库的计算机程序称为 DBMS（数据库管理系统）。例如，SQL Server、Oracle、Sybase、DB2、MySQL、Access 等。当用户想要检索数据库中的数据时，需要通过 SQL 语言发出请求，DBMS 会对该 SQL 请求进行处理并检索所要求的数据，最后将其结果返回给用户，此过程被称作为数据库查询，这也就是数据库查询语言这一名称的由来。SQL 是目前使用最广的、标准的数据库语言。SQL 语言使得在数据库中存取或更新信息变得十分容易。

SQL 不是 C、COBOL 和 FORTRAN 那种完整的计算机语言。SQL 没有用于条件测试的 if 语句，也没有用于程序分支的语句以及循环语句 for 或 do。确切地讲，SQL 是一种数据库子语言，SQL 语句可以被嵌入到另一种语言中，从而使其具有数据库存取功能。SQL 也非严格的结构化语言，它的句法更接近英语语句，很容易理解，大多数 SQL 语句

都是直述其意，读起来就像自然语言一样明了。SQL 还是一种交互式查询语言，允许直接查询存储的数据，利用这一交互特性，可以在很短的时间内解决相当复杂的问题。

本节的内容主要介绍 SQL 的基本语法知识。

6.1.1 select 查询语句

1. 使用 select 语句查询数据

查询是 SQL 语言的核心，而用于表达 SQL 查询的 select 语句则是功能最强也是最为复杂的 SQL 语句，它从数据库中检索数据，并将查询结果提供给用户。

下面通过具体例子说明 select 语句的使用方式。首先建立一个名为 DB 的简单数据库，该库中存放一个叫 S 的用户情况表，如表 6.1 所示。

表 6.1 用户情况表 S

姓名	性别	工资	年龄	电话	居住地区
李一	男	1000	21	1111	上海
吴二	女	2000	22	2222	北京
张三	男	3000	23	3333	成都
李四	女	4000	24	4444	广州
王五	男	5000	25	5555	大连
赵六	女	6000	26	6666	天津
马七	男	7000	27	7777	郑州

在表 6.1 中有六列即六个字段：姓名(xm)、性别(xb)、工资(gz)、年龄(nl)、电话(dh)、居住地区(dz)，下面看看如何用 Select 语句对 S 表中的内容进行查询。

例 6.1 如果要列出 S 表中所有姓名、性别和工资的内容，可用下面的 select 语句

```
select 姓名，性别，工资 from s
```

实现。查询结果如表 6.2 所示。

表 6.2 查询结果

姓名	性别	工资	姓名	性别	工资
李一	男	1000	王五	男	5000
吴二	女	2000	赵六	女	6000
张三	男	3000	马七	男	7000
李四	女	4000			

例 6.2 如果要列出 S 表中所有男性的姓名、性别和工资的内容，可用下面的 select 语句

```
select 姓名, 性别, 工资 from s where 性别="男"
```

实现。查询结果如表 6.3 所示。

表 6.3 有选择的查询结果

姓名	性别	工资	姓名	性别	工资
李一	男	1000	王五	男	5000
张三	男	3000	马七	男	7000

例 6.3 这一次,做一个稍微复杂一点的查询,如果要列出 S 表中年龄大于和等于 24 的所有女性的姓名、工资、电话和居住地区,并且按工资排序,可用下面的 select 语句

```
select 姓名, 工资, 电话 form s where 年龄>=24 and 性别="女" order by 工资
```

实现。查询结果如表 6.4 所示。

表 6.4 复杂的查询结果

姓名	工资	电话	居住地区
李四	4000	4444	广州
赵六	6000	6666	天津

2. 知识点归纳

1) select 语句的四个子句

从这三个例子可以看到,对于简单查询,SQL 的 select 语句和英文语法相似,分析一下它的结构,select 语句可以分为四个子句:select 查询内容、from 查询对象、where 查询条件和 order by 输出方式。

2) select 语句的语法结构

select 语句的语法格式为:

```
select 数据项 1, 数据项 2,… from 表名 where 条件表达式 order by 排序选项
```

select 语句用来从数据库表中检索出满足条件表达式要求的数据项。

3) select 子句的构成

select 子句列出所有要求 select 语句检索的数据表中的内容。它放在 select 语句开始处,指定此查询要检索的数据表中的内容。这些内容通常用数据项表示,即一组用“,”隔开的多个数据项。按照从左到右的顺序,每个数据项产生一列查询结果,一个数据项可能是以下元素:

- 字段名:标识 from 子句指定表中的字段。如果字段名作为数据项,则 SQL 直接从数据库表中每行取出该列的值,再将其放在查询结果的相应行中。
- 常数:指定在查询结果的每行中都放上该值。
- SQL 表达式:说明必须将要放入查询结果中的值按表达式的规定进行计算。

4）from 子句

from 子句列出包含所要查询数据的表，它由关键字 from 后跟一组用逗号分开的表名组成。每个表名都代表一个包括该查询要检索数据的表。这些表称为此 SQL 语句的表源，因为查询结果都源于它们。

5）where 子句

where 子句告诉 SQL 只查询某些行中的数据，这些行用 where 说明的搜索条件进行描述。

6）order by 子句

order by 子句将查询结果按一列或多列中的数据排序。如果省略此子句，则查询结果将是无序的。添加 ASC 属性以升序(从小到大)排列，DESC 属性以降序(从大到小)排列。

7）group by 子句

group by 子句指定汇总查询，即不是对每行产生一个查询结果，而是将相似的行进行分组，再对每组产生一个汇总结果。

8）having 子句

having 子句告诉 SQL 只产生有 group by 得到的某些组的结果，和 where 子句一样，所需要的组也用一个搜索条件来指定。

6.1.2 运算符与函数

SQL 可以进行各种灵活多样的查询，可以使用关系运算符、特殊运算符、逻辑运算符、函数组成条件表达式来进行查询。在这里使用的条件表达式与 JavaScript 中用比较运算符组成的表达式是一致的。

1. 关系运算符

一般通过关系运算符连接一个值来表示查询条件。

(1) ＝　　等于。　　例如，字段名＝“男”。

(2) ＜＞　　不等于。

(3) ＜　　小于。　　例如，字段名＜70。

(4) ＜＝　　小于等于。

(5) ＞　　大于。

(6) ＞＝　　大于等于。　　例如，字段名＞＝20。

2. 特殊运算符

(1) In(字段值列表)　按列表中的值查找，例如，字段名 In("李明","王平","张海")，表示查询该字段中包含“李明”，“王平”，“张海”在内的记录。

(2) Between 初值 and 尾值　例如，字段名 Between 1 and 100，表示查询该字段中 1 到 100 之间的所有数值。

(3) Like　“文本字段的字符”，例如，文本字段名 like “计算机 * ”，表示查询该文本字段中包含“计算机”在内的所有记录。

其中，? 匹配一个字符，* 匹配零或多个字符，用方括号可描述一个可匹配的字符范围。

3. 逻辑运算符

(1) not　例如,姓名 not "李元",表示查询“姓名”字段中除了“李元”的记录。
(2) and　例如,年龄>=24 and 性别="女"。
(3) or　例如,年龄>=30 or 姓名="李元"。

4. 函数

函数有很多,这里仅给出几个日期函数和合计函数的说明。

1) 日期函数

- DAY(date)——给出日期数据中的哪一天的值,例如 DAY(#92-01-01#)为 1。
- month (date) ——给出日期数据中的哪一月的值。
- year (date) ——给出日期数据中的哪一年的值。
- Weekday (date) —— 给出日期数据中星期几的值。
- hour (date) ——给出日期数据中钟点的值。
- date (date) ——给出当前日期。

下面给出一些例子。

- 15 天前的日期可使用表达式:

```
<Date()-15
```

- 20 天之内的日期可使用表达式:

```
Between Date() and Date()-20
```

- 1980 年出生的记录可使用表达式:

```
year([出生日期字段名])=1980
```

- 1999 年 4 月参加工作的记录可使用表达式:

```
year ([参加工作时间])=1999 and month([参加工作时间])=4
```

2) 合计函数

合计函数的说明参见表 6.5。

表 6.5　合计函数

均值	avg(字段名)	对指定字段求平均值
计数	count(字段名)	统计满足条件的记录个数
	count(*)	统计记录(元组)个数
最小	min(字段名)	对指定字段求最小值
最大	max(字段名)	对指定字段求最大值
求和	sum(字段名)	对指定字段求和

例如，select count(ID) from student；用来统计 student 表中按 ID 汇总出的学生人数。

6.1.3 Insert Into 添加语句

SQL 不仅能进行数据库的查询，还可以添加新数据到数据库中。

添加语句的语法格式：

```
Insert Into 表名 (col1, col2…) Values (value1, value2…)
```

Insert Into 添加语句用来向一个表中添加指定的新数据。

例 6.4 将张驰作为一个新的成员加入表 S 中。

```
Insert Into S( xm,xb,nl,gz,dh,dz) values ('张驰', '男', 28, 4500, 8888, '北京')
```

在此语句中，表 S 中列(数据项)的名称列在 S 后面的括号中以逗号隔开，接下去是 Value 短语和括号中同样以逗号隔开的每列数据，应该注意的是数据和列名称的顺序是相同的，而且若是字符串型则以单引号隔开。从概念上来讲，Insert 语句添加的是一个与表列结构相一致的数据行，用取自 Values 子句的数据来填充它，然后将该新行加入表中，表中的行是不排序的，因此不存在将该行插入到表的头或尾或两行之间的这个概念。Insert 语句结束后，新行就是表 S 中的一部分了。

Insert 语句还可以将多行数据添加到另外一个目标表(B)中，在这种形式的 Insert 语句中，添加的数据值由查询语句指定。添加的值来自已经存在的表(A)中。看下面的例子。

例 6.5 如果要把 2008 年 12 月 30 日之前产生的订单编号(Num)、日期(Date)和数目(Amount)从 A 表拷贝到另一个名为 B 的表中去，可用下面的语句：

```
Insert into B (Num,Date,Amount) Select Num,Date,Amount From A Where Date< '30-12-2008'
```

这条语句看起来有些复杂，其实很简单，语句标识了接收新记录的表 B 和接收数据的数据项名称，完全类似于单行 Insert 语句。语句的剩余部分是一个查询，它检索 A 表中的数据。SQL 先执行对 A 表的查询，然后将查询结果逐行插入到 B 表中去。注意表 B 中添加的数据项的格式和表 A 要相同。

6.1.4 Update 更新语句

SQL 还可以对数据库中的数据进行修改和更新，而且更改数据库内容的 SQL 语句更简单。不过，对于一个 DBMS 来说，数据更新所造成的风险大大超出了数据查询。数据库管理系统必须在更改期内保护所存储的数据的一致性，确保有效的数据进入数据库，数据库必须保持一致性，DBMS 还必须协调多用户的并行更新，以确保用户和他们的更改不至于影响其他用户的作业。所以，使用修改和更新语句时，一定要小心！

SQL 用 update 语句更新表中选定行的一列或多列的值。

update 语句的语法格式：

```
update 表名 set 字段名 1=value1 [,字段名 2=value2]… where 条件
```

update 语句使用 value 值更新选定表中指定字段的数据。要更新的目标表在 update 后定义,set 子句指定要更新表中哪些列并指定它们的值。where 语句是不可少的,它用来指定需要更新的行。

例 6.6 这是一个简单的 update 语句,可将表 C 中客户名为 slp 的客户的信贷值更新为 10 万并将他的 ID 变更为 99。

```
update c set credit=100000.00,id=99 where name='slp'
```

例 6.7 如果要将表 C 中客户 ID 为 80、90、100、120 客户的信贷值更新为 20 万,状态值为 021,可用下列语句:

```
update c set credit=200000.00,state=021 where id in (80,90,100,120)
```

其中 where id in (数据集合),表示查询 ID 值为"数据集合"中的哪些行的记录。本例中即搜索 ID 值为 80、90、100、120 的 4 行记录。

SQL 处理 Update 语句的过程是逐行搜索所指定的表,更新满足搜索条件的记录,跳过不满足搜索条件的记录。

6.1.5 Delete 删除语句

Delete 删除语句用来删除数据库表或表中某些行记录。

Delete 删除语句语法格式:

```
Delete from 表名 Where 条件
```

由于 Delete 删除语句过于简单,所以造成的后果是很严重的,虽然其中的 Where 子句是任选的,但它几乎总是存在的,若将 Where 子句从 Delete 语句中省略掉,则被操作的目标表的所有行都将被删除。

例 6.8 从表 D 中删除 ID 为 99 的所有行的记录。

```
Delete from D Where ID=99
```

6.1.6 Create Table 创建表语句

SQL 语言可以分为两大部分:数据定义语言和数据操纵语言,以上介绍的是数据操纵语言的 Select 语句。下面介绍的是数据定义语言,用来创建和修改数据库结构的一种语句,包括 Create 和 Drop 语句。

Create 语句的语法格式:

```
Create Table 表名 (字段名 1 数据类型 (not null),字段名 2 数据类型 (not null),…)
```

Create 语句用来创建一个给定字段的表。

例 6.9 下面的语句创建一个具有字段 name 字符型 30 位, amout 数值型 8 位,ID 数值型 4 位的表 B。

```
Create Table B (name char(30), amout NUMBER(8), id NUMBER(4))
```

虽然 Create Table 比前面介绍的语句难理解一些，但仍然很直观。它将表 B 赋予一个新表，并指定表中三列的名称和数据类型。表被建立后就可以添入数据了。可用下面的语句输入数据。

```
Insert into B(name,amout,id) values('zhangchi',100,1)
```

6.1.7 Drop table 删除表语句

如果不再需要数据库中的某个表，可用 Drop table 语句将该表及其所保存的数据从数据库中删除掉。

Drop 语句的语法格式：

```
Drop table 表名
```

Drop 语句用来删除一个完整的表。

例 6.10 如果要删除表 B 可使用下面的语句。

```
Drop table B
```

好了，SQL 语句就学到这里，可别小看这些貌似英文的简单语句，它们的功能十分强大，使用它们可以对数据库进行操作。

6.2 使用 Access 数据库

数据库从最初的数据文件的简单集合发展到今天的大型数据库管理系统已经成为我们日常生活中不可缺少的组成部分。如果不借助数据库的帮助，许多简单的工作将变得冗长乏味，甚至难以实现。

目前，市面上的数据库产品多种多样，例如 Oracle、SQL Server、MySQL、DB2、Informix、Sybase、Access，从大型企业的解决方案到中小企业或个人用户的小型应用系统，可以满足用户的多样化需求。

Access 是 Microsoft 公司 Office 办公套件中一个极为重要的组成部分，是世界上最流行的桌面数据库管理系统。它提供了大量的工具和向导，即使没有任何编程经验，也可以通过可视化的操作来完成大部分的数据库管理和开发工作。Access 的功能很强大，可以处理公司的客户订单数据，管理自己的个人通讯录，还可以对大量科研数据进行记录和处理。

本节的内容主要介绍如何使用 Access 数据库管理系统创建数据库文件及保存数据的表对象。

6.2.1 创建数据库文件

下面介绍使用 Access 2003 数据库管理软件创建市场调查数据库 questionnaire 的方法。

1. 安装 Access 数据库管理系统软件

确认当前使用的计算机中已经安装了包含 Access 的 Microsoft Office 程序。如果没有安装 Access 数据库管理软件，需要先安装 Access，可使用 Microsoft Office 套件进行安装，其方法与安装其他软件相同。

2. 打开 Access 主窗口

在 Windows 操作系统桌面上选择“开始”→“所有程序”→Microsoft Office Access 2003 命令，启动 Access，打开如图 6.1 所示的 Access 主窗口（如果 Windows 桌面上建立了快捷图标，可以更简单、快捷的启动 Access，只要直接双击桌面上的快捷图标，即可打开 Access 主窗口）。

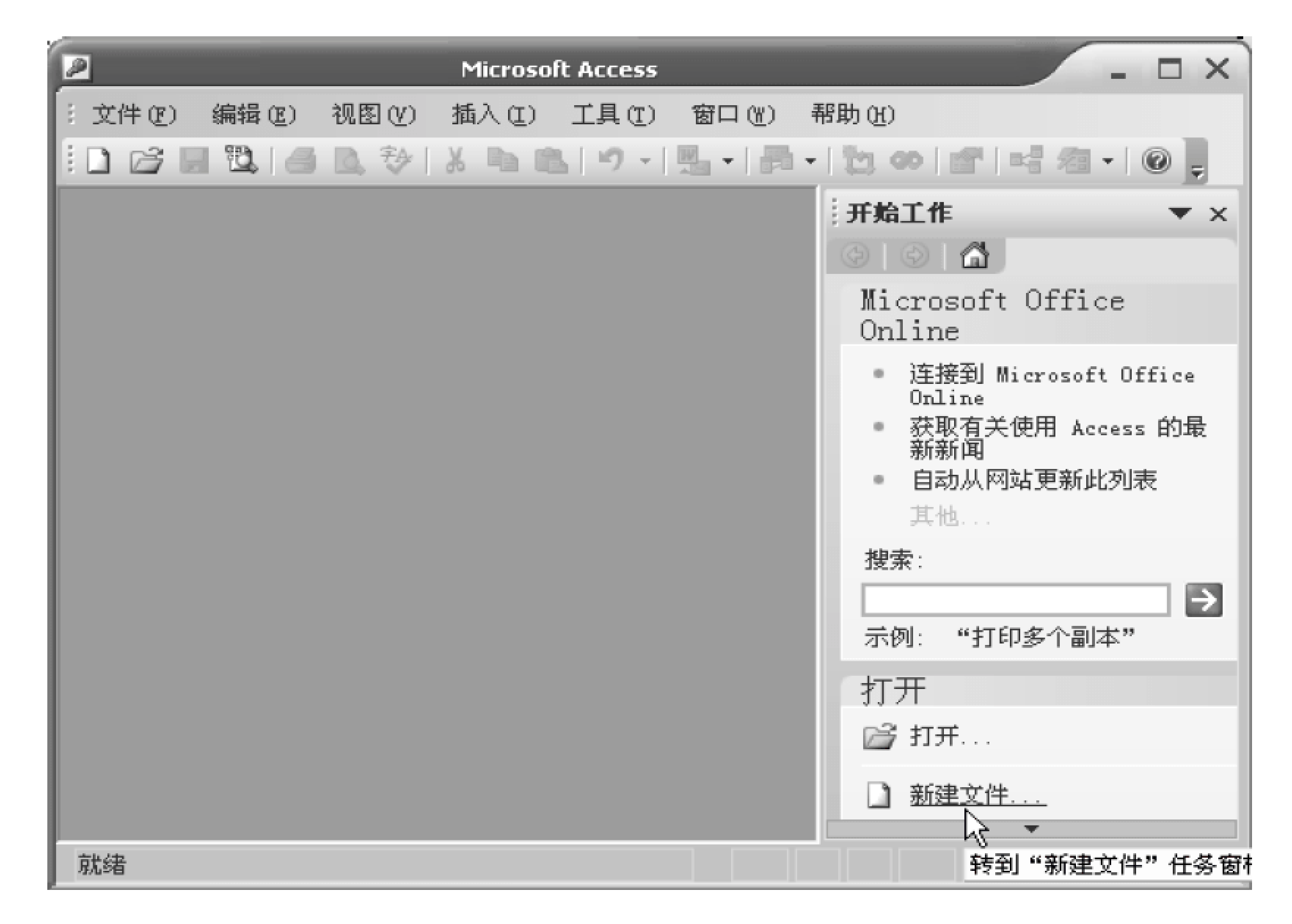


图 6.1 Access 主窗口

3. 查看“开始工作”对话框

首次打开 Access 主窗口时会同时打开“开始工作”对话框，如图 6.1 所示。在对话框中可以根据需要选择不同选项，例如，可在“打开”栏单击“新建文件”菜单，“开始工作”对话框将切换为“新建文件”对话框，如图 6.2 所示。

4. 创建数据库文件

（1）在 Access 主窗口“新建文件”对话框“新建”栏中单击“空数据库”选项，如图 6.2 所示。将打开“文件新建对话框”，如图 6.3 所示。

（2）在“文件新建对话框”中输入数据库文件名称 questionnaire，同时选择文件的保存路径，如图 6.3 所示。

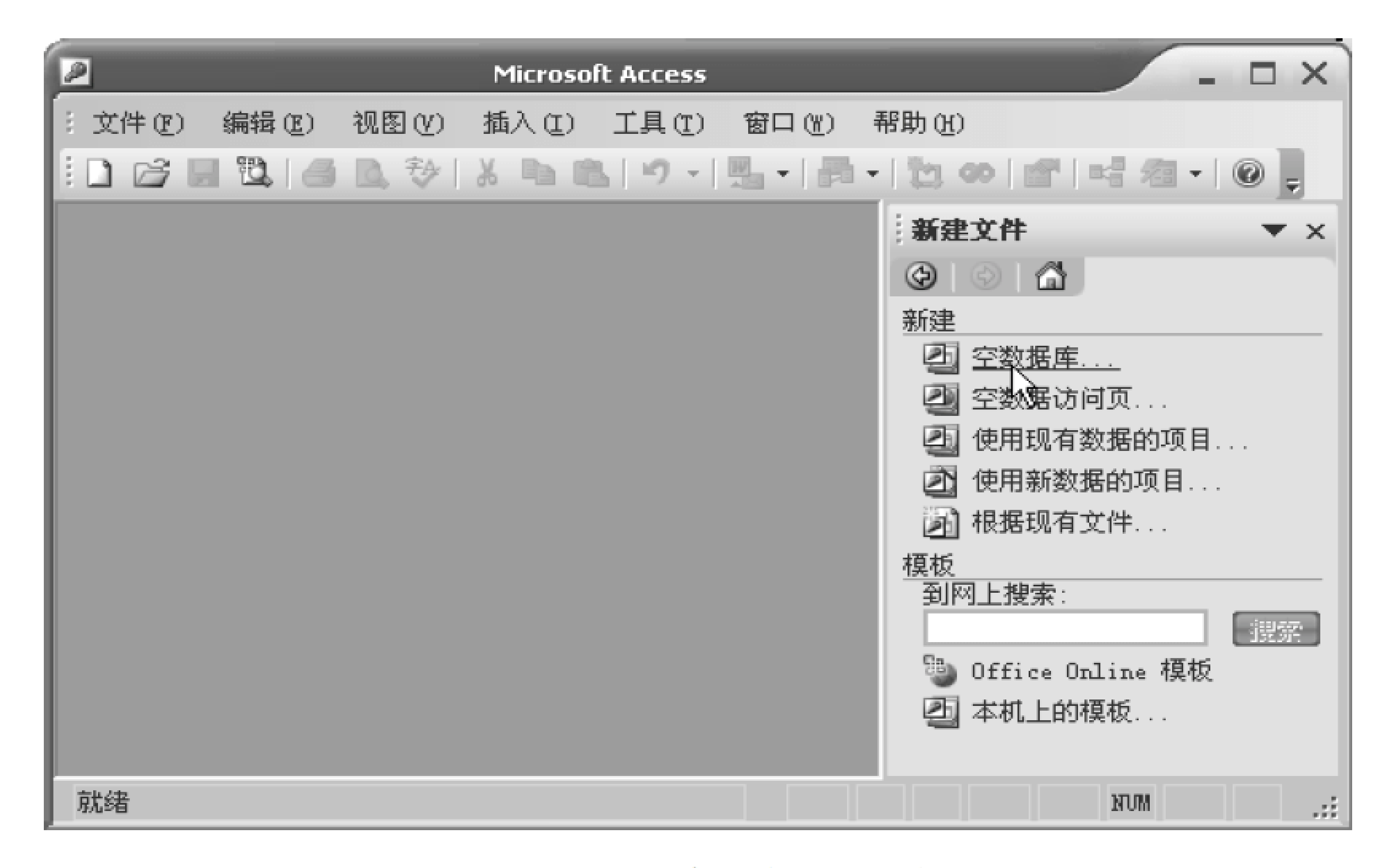


图 6.2 “新建文件”对话框



图 6.3 “文件新建”对话框

(3) 在“文件新建对话框”中单击“创建”按钮,在 Access 主窗口中将打开数据库窗口,如图 6.4 所示,表明已经创建了一个数据库文件 questionnaire。

6.2.2 创建表对象

从本质上说,Access 数据库是一种不断增长的复杂的数据组织结构。Access 数据库由一个或多个数据表组成,一个数据表由一条或多条数据记录组成,一条数据记录由一个或多个数据项(列)组成。

在 Access 数据库中,数据表称为表对象。Access 数据库是存放表对象的一个文件夹,表对象才是真正存放数据的对象。初始创建的数据库文件只是一个空的框架,里面没有存放任何数据。要存放数据,需要在数据库中创建存放数据的表对象。所以,建立 Access 数据库的主要工作就是创建表对象。

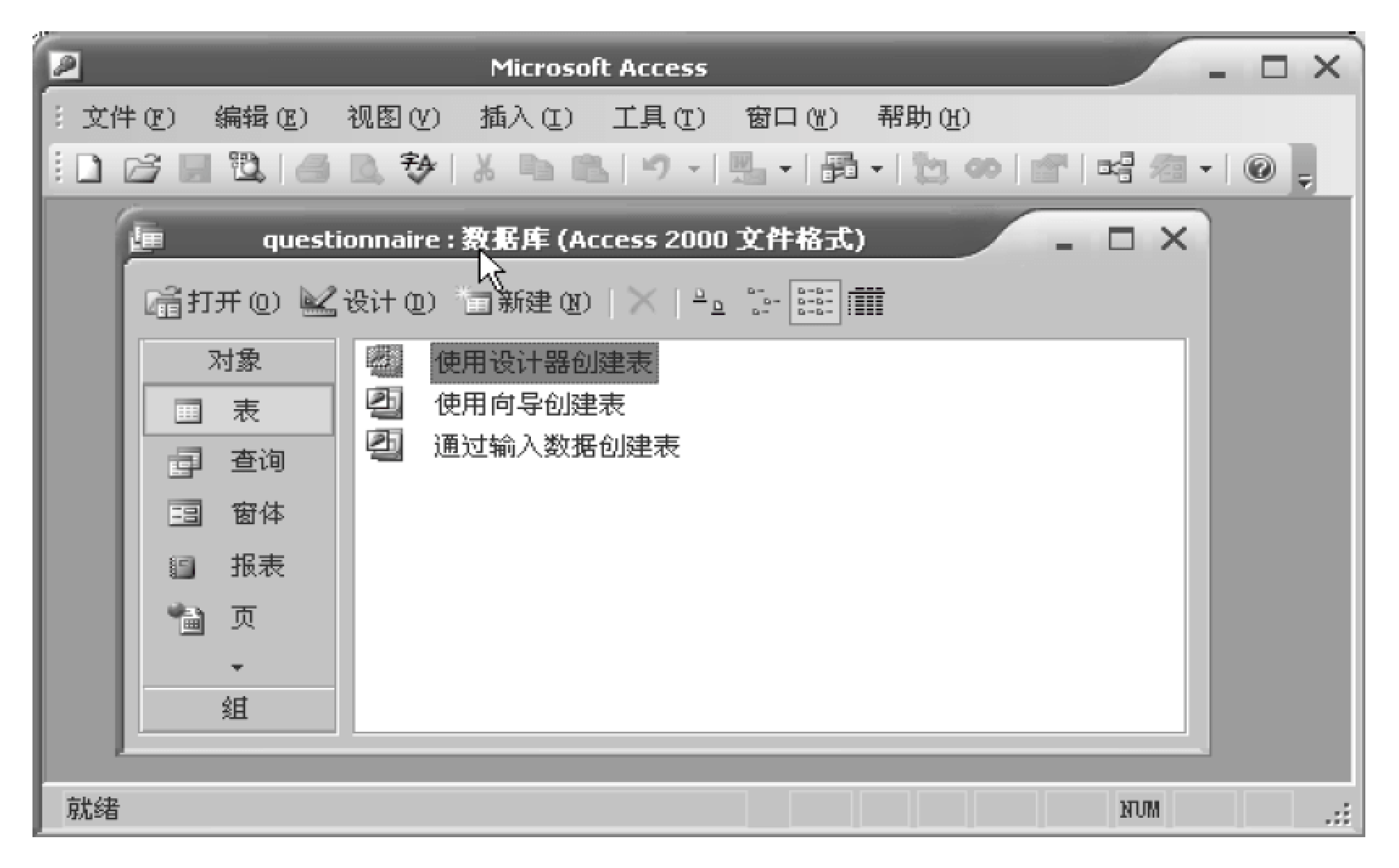


图 6.4　questionnaire 数据库窗口

下面介绍如何在 questionnaire 数据库中创建管理员密码表对象 admins、调查主题表对象 questions、调查选项表对象 items。

1. 创建调查主题表 questions 表结构

表对象由表结构与表数据构成。表结构指表名、字段名、数据类型、字段大小等。创建表结构后才能在其中存放表数据。如同先有房子后住人一样。

创建表结构可以分为设计逻辑表结构与创建物理表结构两个过程。设计逻辑表结构即定义表名、字段名、字段数据类型与字段属性等,这些工作一般在数据库设计阶段时完成。

创建物理表结构即在计算机中存放的数据库文件中定义表名、字段名、字段数据类型与字段大小等,其结果是为存放的数据在计算机中准备存储空间,下面介绍的创建表结构是指创建物理表结构。

创建物理表结构的操作步骤如下所示。

(1) 在 questionnaire 数据库窗口“对象”栏按下“表”按钮,单击数据库窗口工具栏上的“设计”按钮或双击“使用设计器创建表”创建方法选项,打开如图 6.5 所示的表设计视图窗口。



图 6.5　表设计视图窗口

（2）单击表设计视图第 1 行“字段名称”单元格，输入第 1 个字段名称 questionID。

（3）在“数据类型”单元格，单击其右边的向下箭头按钮，在其下拉列表中列出 Access 的所有数据类型，选择“自动编号”，在“说明”单元格中输入“主关键字”，如图 6.6 所示。



图 6.6　定义字段名称、类型

（4）输入字段名称 question，选择“数据类型”为“文本”型，在“字段属性”区“常规”标签下“字段大小”栏输入 255，如图 6.7 所示。



图 6.7　定义字段名称、类型、大小

（5）输入字段名称 Isvisable，选择数据类型为“数字”型，在“字段属性”区“常规”标签下“字段大小”栏选择“字节”，在“默认值”栏输入 0，如图 6.8 所示。

（6）输入字段名称 Date，选择数据类型为“日期/时间”型，在“字段属性”区“常规”标签下“格式”栏选择“常规日期”，如图 6.9 所示。

（7）输入字段名称 IsOpen，选择数据类型为“数字”型，在“字段属性”区“常规”标签下“字段大小”栏选择“字节”，在“默认值”栏输入 1，如图 6.10 所示。



图 6.8　定义字段名称、类型、大小、默认值



图 6.9　定义字段名称、类型、格式



图 6.10　创建表结构

2. 定义主关键字

将表中所有的字段定义完毕后，选择最右边的字段选择器按钮，选择 QuestionID 字段，然后单击主窗口工具栏的“主键”按钮或单击右键在弹出菜单选择“主键”命令，将设置该字段为主关键字，如图 6.11 所示。

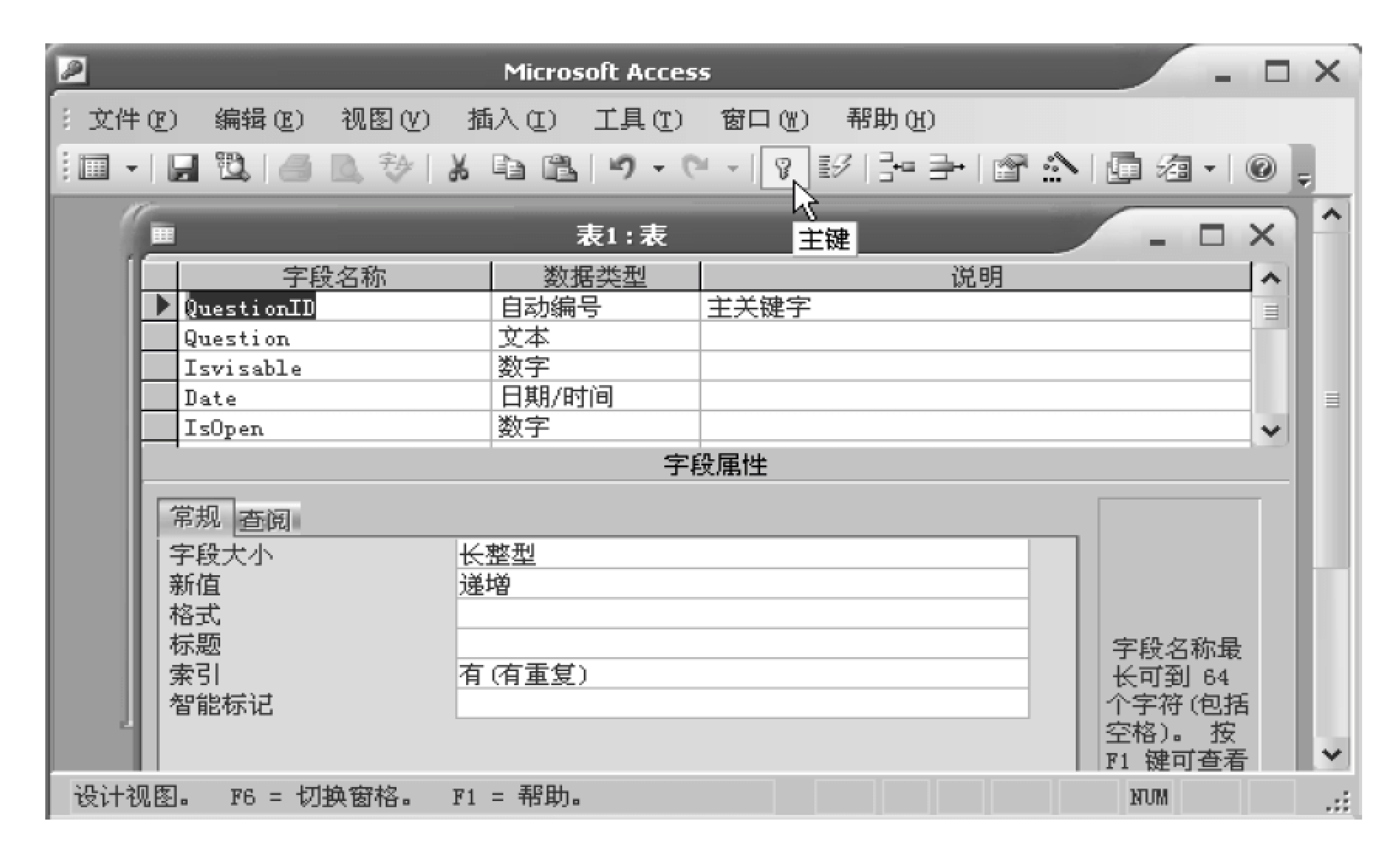


图 6.11 定义关键字

设置主键后，该字段行选择器按钮上会出现一个小钥匙，如图 6.12 所示。

表1 : 表

字段名称	数据类型	说明
QuestionID	自动编号	主关键字
Question	文本	
Isvisable	数字	
Date	日期/时间	
IsOpen	数字	

图 6.12 关键字符号

如果主键由多个字段组成，可按住 Ctrl 键不放，然后单击字段选择器选中每个作为主键的字段，再单击“主键”按钮，即可同时将多个字段同时设置为主键了。

如果要取消字段的主键定义，在选择主键字段后单击“主键”按钮即可。

主键字段(关键字字段)是唯一标识记录的字段。

3. 以表名 questions 保存调查主题表对象的表结构

用表设计器设计好表结构之后可关闭表设计视图并保存表结构。

(1) 单击工具栏上的“保存”按钮，会弹出如图 6.13 所示对话框。

(2) 单击“是”按钮，弹出如图 6.14 所示“另存为”对话框，在“表名称”框中输入 questions 并单击“确定”按钮，即完成表结构的创建工作。

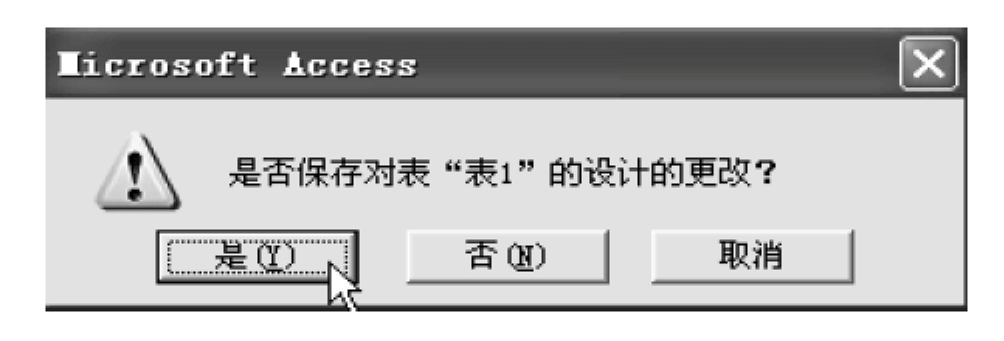


图 6.13　保存表结构对话框



图 6.14　选取表名称对话框

(3) 单击表设计视图右上角关闭按钮☒,可关闭表视图,结束创建表的任务,在数据库窗口可以看到刚刚创建的表对象,如图 6.15 所示。

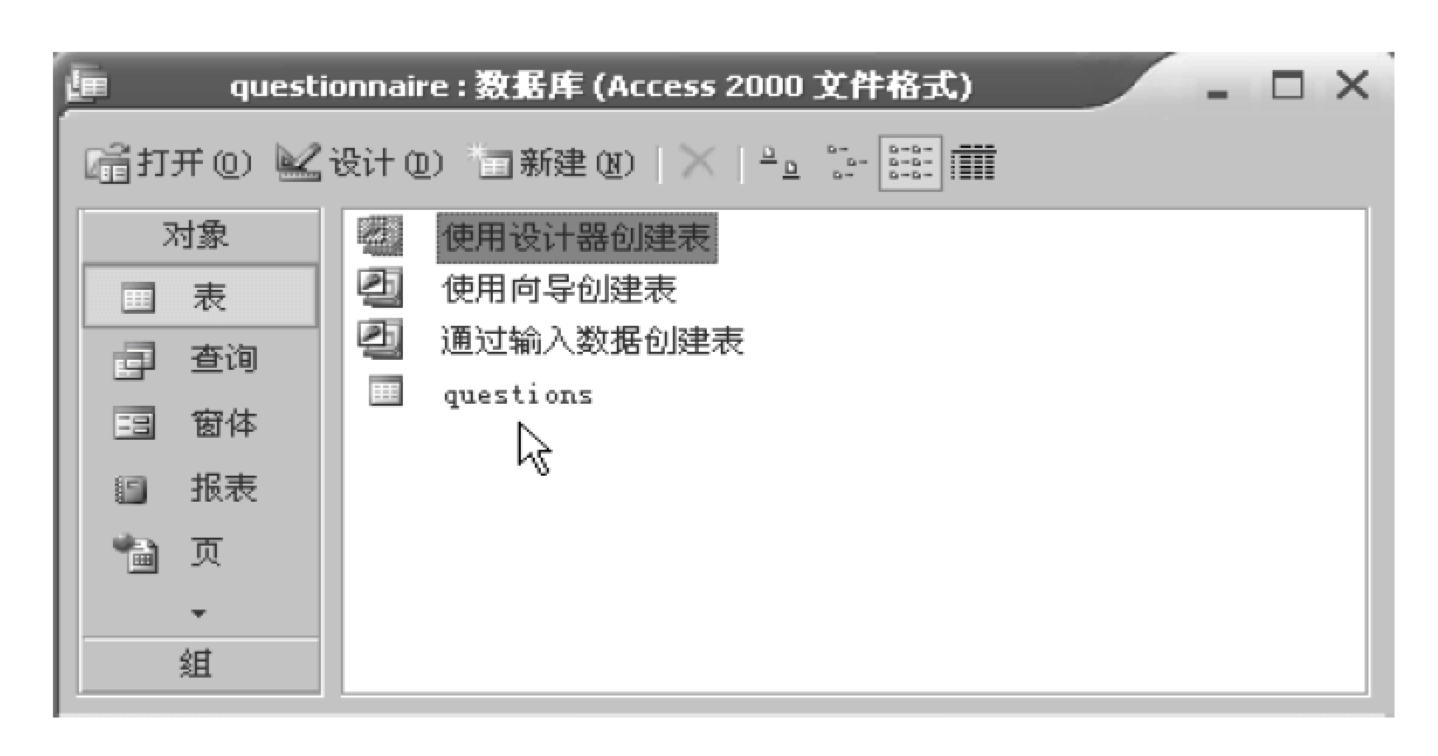


图 6.15　创建的表对象

类似方法可以创建管理员密码表对象 admins、调查选项表对象 items 的表结构。

6.2.3　在表对象中添加表数据

1. 以数据表视图打开表对象

(1) 启动 Access,在"开始工作"对话框"打开"栏单击已经创建的数据库 questionnaire,再次打开 questionnaire 数据库窗口。

(2) 在数据库窗口"对象"栏中按下"表"按钮,在"已有对象列表"中双击 admins 表名称或选中 admins 表对象单击数据库窗口工具栏"打开"按钮,将以数据表视图打开如图 6.16 所示的 admins 表结构或称为空表。



图 6.16　空表 admins

(3) 在 admname 单元格输入管理员名称 lpshao、zhc,在 admpwd 输入口令 123456、111111,如图 6.17 所示。



图 6.17　向 admins 表输入的数据

类似方式可向其他表中添加数据。

2. 切换不同视图窗口

在数据表视图窗口单击主窗口工具栏的“视图”按钮，可以切换到表设计视图窗口，可以修改表结构。在表设计视图窗口单击主窗口工具栏的“视图”按钮，可以切换到数据表视图窗口在表中添加数据。

6.3　ODBC 与 JDBC

如何在 JSP 应用程序中使用数据库(questionnaire)中的数据呢？通过 JSP 访问数据库中的数据还需要搭建 ODBC 桥梁并使用 JDBC 作为交通工具。

本节的内容主要介绍搭建 ODBC 桥梁的方法与 JDBC 的作用。

6.3.1　搭建 ODBC 桥梁

1. 什么是 ODBC

ODBC 是英文 Open Database Connectivity 的缩写，中文含义为开放式数据库互联。ODBC 是微软推出的一种工业标准，一种开放的独立于厂商的 API 应用程序接口，可以跨平台访问各种个人计算机、小型机以及主机系统。

ODBC 是一种访问数据库的工具，只要操作系统中有相应的 ODBC 驱动程序，任何程序都可以通过 ODBC 来操纵数据库。比如操作系统中如果有 Access 的 ODBC 驱动程序，那么即使计算机中没有 Access 软件，也可以在 JSP 应用程序中对 Access 数据库文件进行添加、删除、更新记录的操作，而且不必知道这个数据库文件的存放位置。只要写出 SQL 指令，ODBC 驱动程序会解决访问数据库文件的一切问题。

2. 定义数据源名称 DSN

DSN 是英文 Date Source Name 的缩写，中文名称为数据源名称。它用来定位和标识 ODBC 兼容的数据库。DSN 是 Web 应用程序和数据库之间连接的纽带，搭建 ODBC 桥梁有两个任务，一个是选择数据库驱动程序。再一个重要任务就是设置 DSN。

在 Windows XP 操作系统中自带有“ODBC 数据源管理器”专门用来选择数据库驱动程序与定义 DSN。定义 DSN 后，即可将 SQL 指令通过 Web 应用程序传递给数据库，对数据库进行各种操作。因为搭建 ODBC 桥梁的标志为定义出 DSN，所以，定义好 DSN 即完成了搭建 ODBC 桥梁的工作。

下面介绍定义连接 questionnaire 数据库的数据源 jsp 的方法，操作步骤如下所述。

(1) 在 Windows 界面选择 “开始”→“控制面板”命令，打开“控制面板”窗口，如图 6.18 所示。



图 6.18 控制面板对话框

(2) 在“控制面板”对话框中双击“管理工具”图标，打开“管理工具”对话框，如图 6.19 所示。

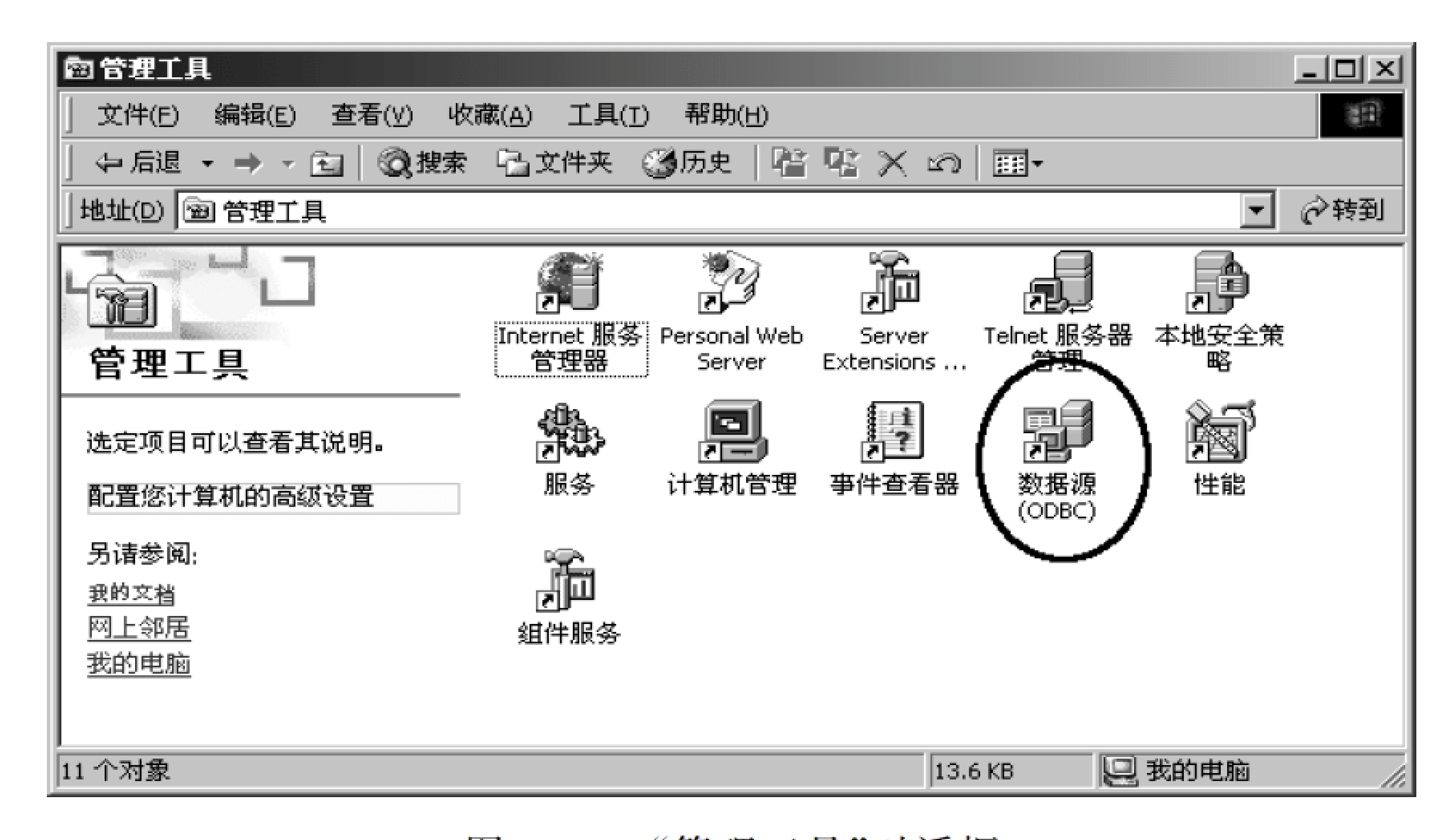


图 6.19 “管理工具”对话框

(3) 在“管理工具”对话框中双击“数据源(ODBC)”图标，打开“ODBC 数据源管理器”对话框，如图 6.20 所示，选择“系统 DSN”标签。

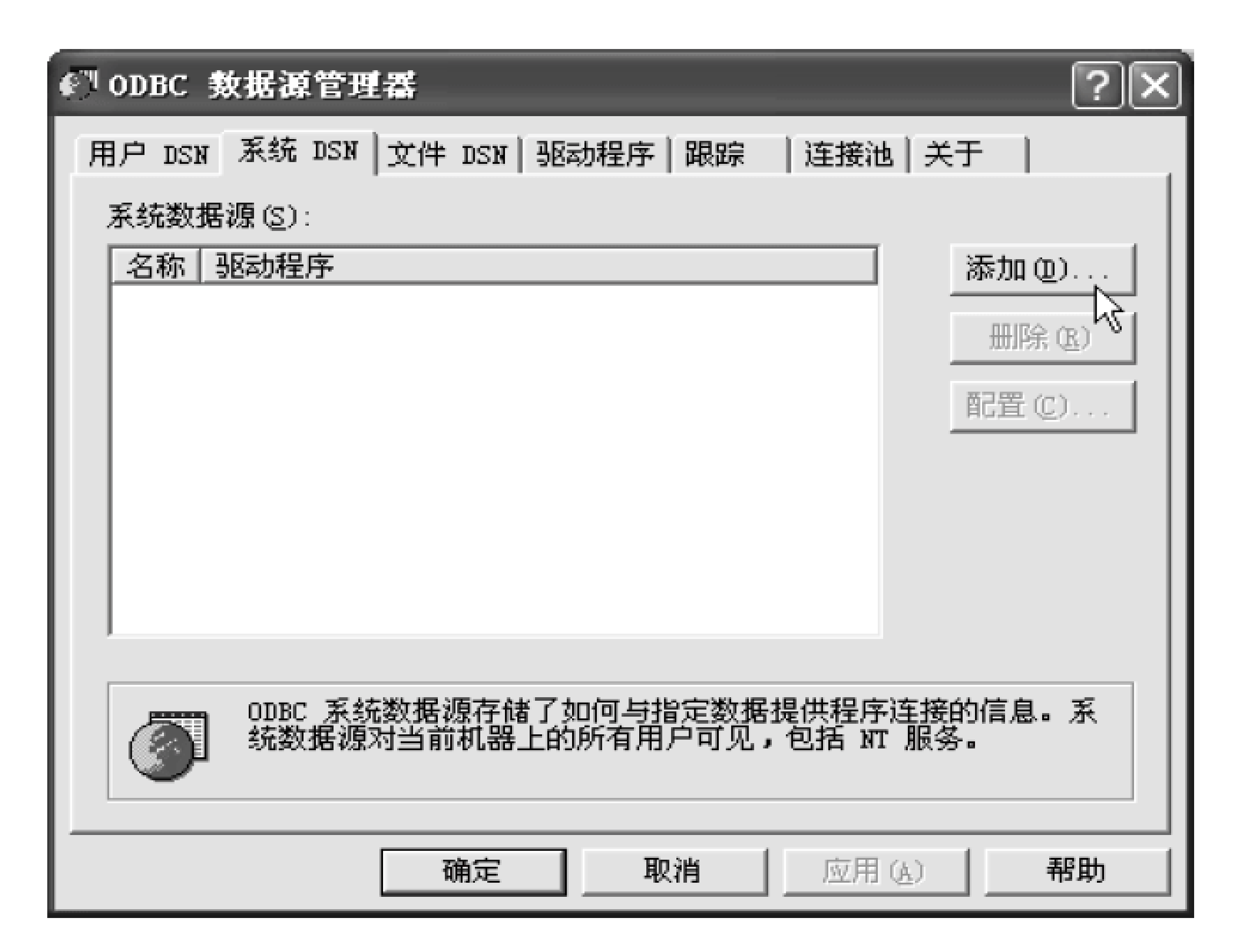


图 6.20 “ODBC 数据源管理器”对话框

(4) 在图 6.20 中单击“添加”按钮，将弹出“创建新数据源”对话框，如图 6.21 所示。选择 Microsoft Access Drive 选项并单击“完成”按钮。将弹出“ODBC Microsoft Access 安装”对话框，如图 6.22 所示。



图 6.21 “创建新数据源”对话框

(5) 在图 6.21 的“数据源”栏中输入 DSN 名称 jsp，然后单击“数据库”栏中“数据库:”下的“选择”按钮，将打开“选择数据库”对话框，如图 6.23 所示。

(6) 在“选择数据库”对话框中先在“目录”框下选择数据库存放的路径，再选择要使用的数据库，例如“c:\WZKF\myjsp\DBA\questionnaire.mdb”，然后单击“确定”按钮返回“ODBC Microsoft Access 安装”对话框。

(7) 在“ODBC Microsoft Access 安装”对话框中可看到数据库文件的路径及名称如图 6.24 所示，单击“确定”按钮返回“ODBC 数据源管理器”对话框。

图 6.22 “ODBC Microsoft Access 安装”对话框



图 6.23 “选定数据库”对话框



图 6.24 “ODBC Microsoft Access 安装”对话框

(8) 在“ODBC 数据源管理器”对话框可看到创建的系统数据源 jsp，如图 6.25 所示，单击“确定”按钮就完成了定义 DSN 的工作。

完成以上操作步骤就建立了一个连接数据库 questionnaire.mdb 的数据源 jsp，即搭建了一个 ODBC 桥梁。JSP 程序可以通过数据源 jsp 找到存放在 questionnaire.mdb 数据库中的表对象以及其中的数据。如果数据库改变了名称或存放路径，不必一个一个地修改程序中所有数据库的名称，只要修改 DSN 配置中数据库的存放路径就可以了。

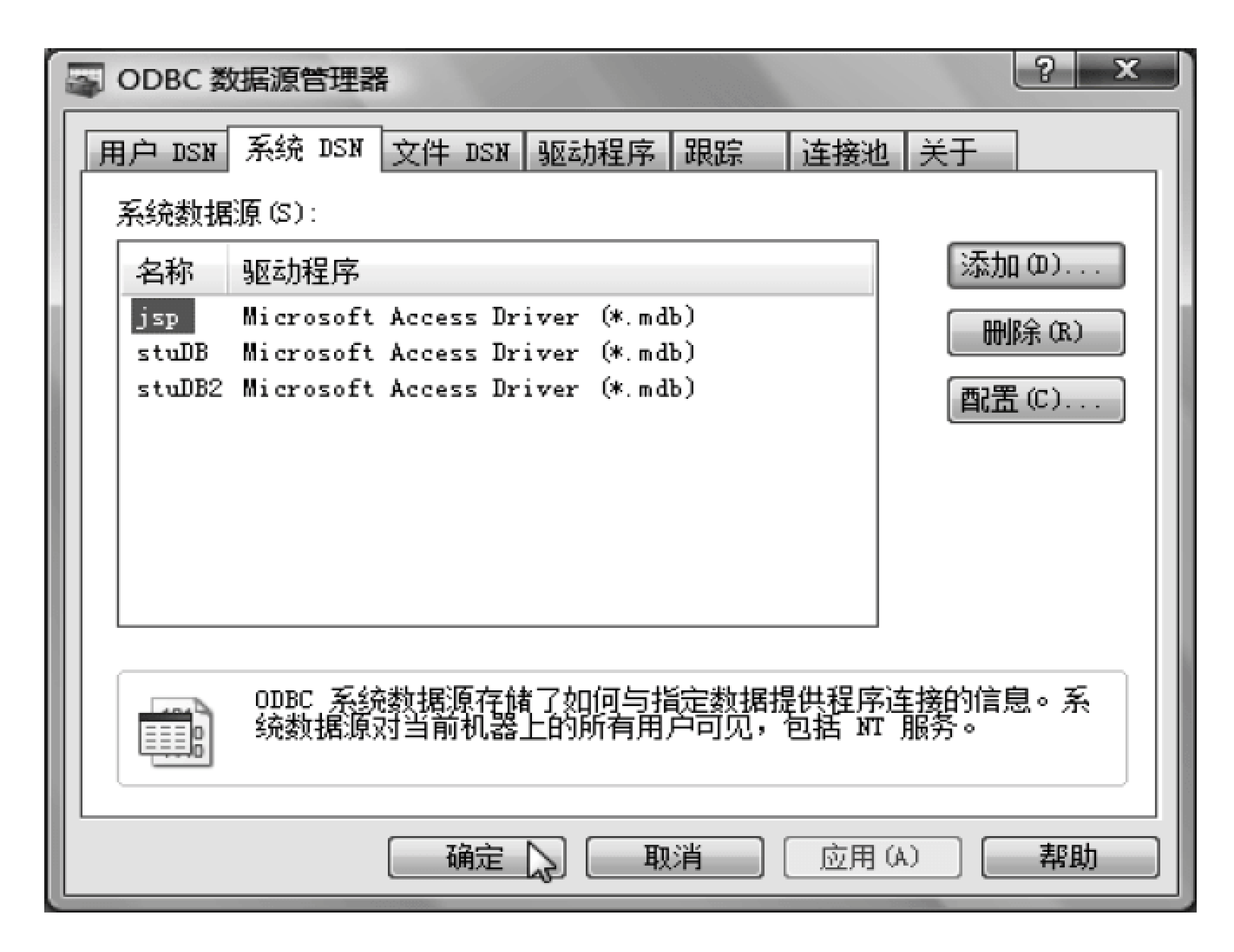


图 6.25 “ODBC 数据源管理器”对话框

6.3.2 JDBC 的作用

如何在 JSP 文件中使用数据源 jsp 连接的数据库中的数据呢？还需要 JDBC 作为交通工具。

1. 什么是 JDBC

JDBC 是英文 Java Data Base Connectivity 的缩写，中文称为 Java 数据库连接。它是 Sun 提供的标准数据库应用编程接口（Application Programming Interface，API），它由 Java 语言编写的类和接口组成，存放在 java.sql 类包中，目的是方便开发人员编写数据库应用程序。

JDBC 可以看成是一种访问数据库的方法，只要系统中有相应的 JDBC 驱动程序，Java 应用程序就可以通过 JDBC 操纵数据库，对数据库进行加、删、改记录的操作。而且你根本不用知道这个数据库放在哪里。

2. 常用的 JDBC 接口

（1）java.sql.DriverManager：处理驱动程序的调入，对产生新的数据库连接提供支持。

（2）java.sql.Connection：用于对特定数据库进行连接。

（3）java.sql.Statement：代表一个特定的容器，用来对一个特定的数据库执行 SQL 语句。

（4）java.sql.ResultSet：用于控制对一个特定语句行数据的存取。

关于 java.sql 包的接口及其方法可到网址 http://gceclub.sun.com.cn/Java_Docs/html/zh_CN/api/index.html 详细查看。

3. JDBC 的功能

JDBC 的实现类称为 JDBC 驱动程序。JDBC 驱动程序具有以下功能：

(1) 同一个数据库建立连接；

(2) 向数据库发送 SQL 语句；

(3) 返回数据库处理的结果。

这些功能是通过 JDBC 中的一系列接口来实现的，这些接口都在 java.sql 包中。所以，在编写访问数据库的 JSP 程序时一定要包含引入 java.sql 包的 Java 语句。

6.4 在服务器端通过 JSP 访问数据库

本节的内容将通过具体实例介绍在服务器端通过 JSP 文件访问数据库的不同方式。

6.4.1 通过 JSP 页面显示数据库中的数据

1. 查询数据库中的数据

例 6.11 查询并使用表格显示数据库中数据的程序(6-11.JSP)。

```
<%@page contentType="text/html;charset=gb2312"%>
<%@page import="java.sql.*"%>
<html>
<body>
<b>下面显示的是数据库中的数据!</b><p>
<%
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");            //注册驱动程序
Connection c=DriverManager.getConnection("jdbc:odbc:jsp");
                                                          //创建连接数据源的对象
Statement s=(Statement)c.createStatement();
String sql="select * from admins";
ResultSet rs=s.executeQuery(sql);
out.println("<table border=1>");
while(rs.next()) {%>
<%out.println("<tr><td>");
out.println(rs.getString(1));
out.println("</td>");
out.println("<td>");
out.print(rs.getString(2));
out.println("</td></tr>");%>
<%}%>
<%rs.close();
s.close();
c.close();%>
```

```
</body>
</html>
```

在浏览器中输入 http://localhost:8080/myjsp/6-11.jsp，运行结果如图 6.26 所示。可以看到在页面上显示了数据库 questionnaire 中表对象 admins 中的数据。



图 6.26　在 JSP 页面中显示数据库中的数据

2. 知识点归纳

从例 6.11 中可以发现，要使用 SQL 语句对数据库中数据的数据进行查询，在 JSP 文件中要注意以下事项：

1）引入 SQL 类包

必须使用<%@ page import="java.sql.*"%>语句引入 Java 的 SQL 类包，因为 JDBC 中的一系列接口程序都在 java.sql 包中，要访问数据库，必须使用 JDBC 中的接口程序。

2）装载并注册驱动程序

连接 JdbcOdbc 驱动程序的类名为 sun.jdbc.odbc.JdbcOdbcDriver，使用驱动程序类，要使用 Class 类的静态方法 forName 获取驱动程序对象。一般使用下面的语句装载并注册驱动程序：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

3）创建与数据库建立连接的 Connection 对象

Connection 对象来自于 java.sql.Connection 接口，它的作用是与数据库进行连接，通过 DriverManager 类的 getConnection(url)方法，可以创建一个 Connection 对象，可用以下语句创建 Connection 对象：

```
Connection c=DriverManager.getConnection(url);
```

4）URL 地址

要使用 JDBC 连接数据库需要建立一个 URL 地址，可以使应用程序知道到哪里找到数据库中的表。例 6.11 中建立 URL 地址为"jdbc:odbc:jsp"。

创建 URL 地址的语法格式如下：

```
url=jdbc:odbc://[hostname][:port]/dbname[?user=value1][&password=value2] [&param1=
value3] [&param2=value4]…
```

其中可包括如下参数：

- hostname，主机名称；
- port，端口号；
- dbname，数据库名称，不能缺少；
- user，数据库的用户名；
- password，连接 MySQL 数据库的用户口令；
- autoReconnect，当数据库连接丢失时是否自动连接，取值 true/false，默认时为 false；
- maxReconnects，如果 autoReconnect 为 true，此参数为重试次数，默认为 3 次；
- initialTimeout，如果 autoReconnect 为 true，此参数为重新连接前等待的秒数，默认为 2 秒；
- maxRows，设置查询时返回的行数，0 表示全部，默认为 0；
- useUnicode，是否使用 unicode 输出，true/false，默认为 false；
- characterEncoding，如果 useUnicode 为 true，该参数制定 encoding 类型，建议使用 gb2312，默认为无。同时使用 useUnicode，characterEncoding，能解决数据库输出时的中文问题。

5）创建执行 SQL 语句的 Statement 对象

执行 SQL 语句的 Statement 对象来自于 java. sql. Statement 接口，它的作用是对一个特定的数据库执行 SQL 语句操作。通过 Connection 对象的 createStatement()方法经过 Statement 类型转换可以得到一个 Statement 对象，例如下面的语句：

```
stmt= (Statement)c.createStatement();
```

Statement 对象可以对多个不同的 SQL 语句进行操作。

6）ResultSet 对象

ResultSet 对象来自 java. sql. ResultSet 接口，它被称为结果集，代表一个特定的容器，用来保存查询的所有结果数据。ResultSet 对象是通过 Statement 对象的 executeQuery(sql)方法在执行 SQL 语句后创建的，例如下面的语句：

```
ResultSet rs=s.executeQuery(sql);
```

ResultSet 对象可以根据查询结果按行对数据进行存取。存取数据时可用到以下方法：

next()，可以移动指针到查询到的当前数据行的下一行。

getXXXX(n)，可以给出查询到的当前行数据第 n 列的数值。XXXX 表示不同的数据类型，例如 getLong(1)，getString(2)。关于 java. sql 包的接口及其方法可到网站详细查看。

7）在页面上用表格显示数据库中的数据

在例 6.11 中使用 HTML 表格标记码与 JSP 表达式语句、Java 的 while 循环语句在

页面上使用表格显示数据库表对象中的数据。

8）释放资源

最后要使用 close()方法释放所定义的 s、c、rs 对象资源。

6.4.2 通过 JSP 文件在数据库中添加与删除数据

1. 在数据库中添加与删除数据

例 6.12 在数据库中添加与删除数据的程序(6-12.JSP)。

```
<%@page contentType="text/html;charset=gb2312"%>
<%@page import="java.sql.*"%>
<html>
<body>
<b>显示添加与删除后的数据</b><p>
<%
PreparedStatement ps;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");            //注册驱动程序
Connection c=DriverManager.getConnection("jdbc:odbc:jsp");
                                                          //创建连接数据源的对象
Statement s=(Statement)c.createStatement();
//添加记录
String sql1="insert into admins values(?,?)";
ps=(PreparedStatement)c.prepareStatement(sql1);
        ps.setString(1,"张驰");
        ps.setString(2,"888888");
        ps.executeUpdate();
//删除记录
String sql2="delete from admins where admname='zhc'";
        s.executeUpdate(sql2);
//查询记录
ResultSet rs=s.executeQuery("select * from admins");
out.println("<table border=1>");
while(rs.next()) {
out.println("<tr><td>");
out.println(rs.getString(1));
out.println("</td>");
out.println("<td>");
out.print(rs.getString(2));
out.println("</td></tr>");
}
rs.close();
s.close();
c.close();%>
</body>
```

```
</html>
```

在浏览器中输入 http://localhost:8080/myjsp/6-12.jsp，运行结果如图 6.27 所示。可以看到在页面上显示了添加到数据库 questionnaire 中表对象 admins 中的数据“张驰，888888”。

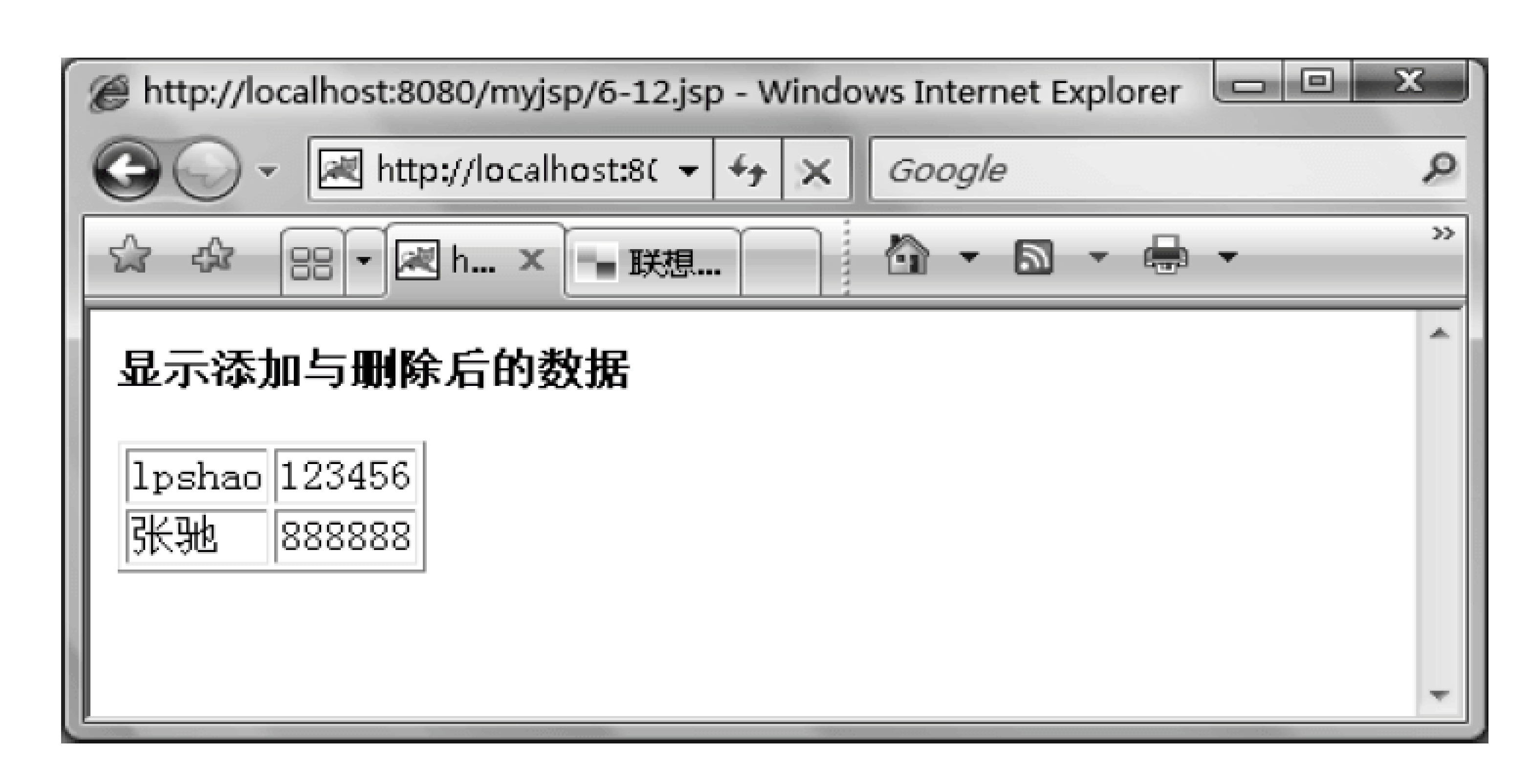


图 6.27　6-12.jsp 的运行结果

注意：如果再次运行 6-12.jsp 会发生添加数据出错的提示，因为姓名字段是关键字，不能重复。

2. 知识点归纳

(1) PreparedStatement 对象

例 6.12 中使用了 Statement 的子接口 PreparedStatement 对象，它的功能更强大，它使用的 SQL 语句中可以包含用多个问号“?”代表的字段，这样的 SQL 语句称为预编译的 SQL 语句，例如

```
String sql="insert into admins values(?,?)";
```

通过 PreparedStatement 对象的 setXXXX()方法可以分别给“?”代表的字段赋值，例如

ps.setString(1, "文本数据")；添加文本类型数据，使用 ps.setInt(2, 12) 添加整数型数据，按 SQL 语句中“?”字段出现的顺序为记录中的字段添加数据，从 1 开始。

通过 PreparedStatement 对象的 executeUpdate()方法完成添加数据的任务。

(2) 使用不同的 SQL 语句

通过例 6.12 可以看到，连接数据库的方式是相同的，不同的是 SQL 语句与执行 SQL 语句的对象。

6.5　在客户端通过页面访问数据库

在 6.4 节中介绍了在服务器端通过 JSP 文件访问数据库的方式，但这种处理方式不灵活，需要在服务器端的程序文件中编写相应的代码才能解决查询、添加、删除等任务，能

不能在客户端通过页面直接向数据库添加数据或进行查询等操作呢?

本节的内容主要解决在客户端通过页面访问数据库的任务。

6.5.1 创建两个重复使用的共用文件

为了使 JSP 代码简单清晰,可将连接数据库的任务单独编写为一个共用的 JSP 文件,将数据库查询结果按表格输出的任务编写为一个共用的 JSP 文件。当其他程序需要使用数据库连接或用表格输出数据库查询的结果时,只要稍加修改或直接将共用的 JSP 文件包含在 jsp 文件中即可。这样可以使代码重复使用,还可以增加程序的可维护性。

1. 创建完成连接数据库任务的 JSP 文件

例 6.13 连接数据源 jsp 的 JSP 文件(jdbc-jsp. jsp)。

jdbc-jsp. jsp 源代码如下:

```
<%@page import="java.sql.*"%>
<%
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String url="jdbc:odbc:driver={Microsoft Access Driver (*.mdb)}; "+
"DBQ=C://WZKF//myjsp//DBA\\questionnaire.mdb";
Connection c=DriverManager.getConnection(url);
Statement s=(Statement)c.createStatement();
PreparedStatement ps=null;
%>
```

jdbc-jsp. jsp 解决了连接数据库 questionnaire 的任务,创建了一个连接数据库对象 c、一个进行 SQL 查询的容器对象 s 和一个进行 SQL 添加操作的对象 ps。并提供了一种直接与数据库文件建立联系的方法。可以免去建立数据源的工作。其中,DBQ 用来说明数据库所在的物理目录,例如本例中,DBQ 表示数据库存放在 C:/WZKF/myjsp/DBA/questionnaire. mdb 路径下。

2. 创建完成用表格输出数据库查询任务的 JSP 文件

例 6.14 按表格输出数据库查询结果的 JSP 文件(table. jsp)。

table. jsp 源代码如下:

```
<%
out.println("<table border=1>");
while(rs.next()) {
out.println("<tr><td>");
out.println(rs.getString(1));
out.println("</td>");
out.println("<td>");
out.print(rs.getString(2));
out.println("</td></tr>");
```

```
}
rs.close();
s.close();
c.close();
%>
```

6.5.2 在客户端向数据库添加数据

1. 创建用户输入数据页面

例 6.15 创建一个在客户端浏览器中用户输入数据的 HTML 页面(6-16.htm)。

```
<strong>向数据库添加数据</strong>
<form name="form1" method="post" action="6-16.jsp">
<p>用户名<input name="yhm" type="text" id="yhm" size="15"></p>
<p>口  令<input type="password" name="kl" size="15"></p>
<p><input type="submit" name="Submit" value="提交">
<input type="reset" name="" value="重置"></font></p>
</form>
```

2. 创建数据处理与显示数据结果的页面

例 6.16 向数据库添加数据并显示添加结果的 JSP 页面(6-16.jsp)。

6-16.jsp 源代码如下:

```
<%@page contentType="text/html;charset=gb2312"%>
<b>在客户端添加数据到数据库</b><p>
<%@include file="jdbc-jsp.jsp" %>
<%
//添加数据
String yhm=request.getParameter("yhm");
String kl=request.getParameter("kl");
ps=c.prepareStatement("insert into admins values(?,?)");
ps.setString(1,yhm);
ps.setString(2,kl);
ps.execute();
//查询记录
ps=c.prepareStatement("select * from admins ");
ResultSet rs=ps.executeQuery();
%>
<%@include file="table.jsp" %>
```

在浏览器中输入 http://localhost:8080/myjsp/6-16.htm,将显示如图 6.28 所示的输入数据页面,当数据输入后按"提交"按钮后可看到如图 6.29 所示的显示数据页面。



图 6.28　用户输入数据页面

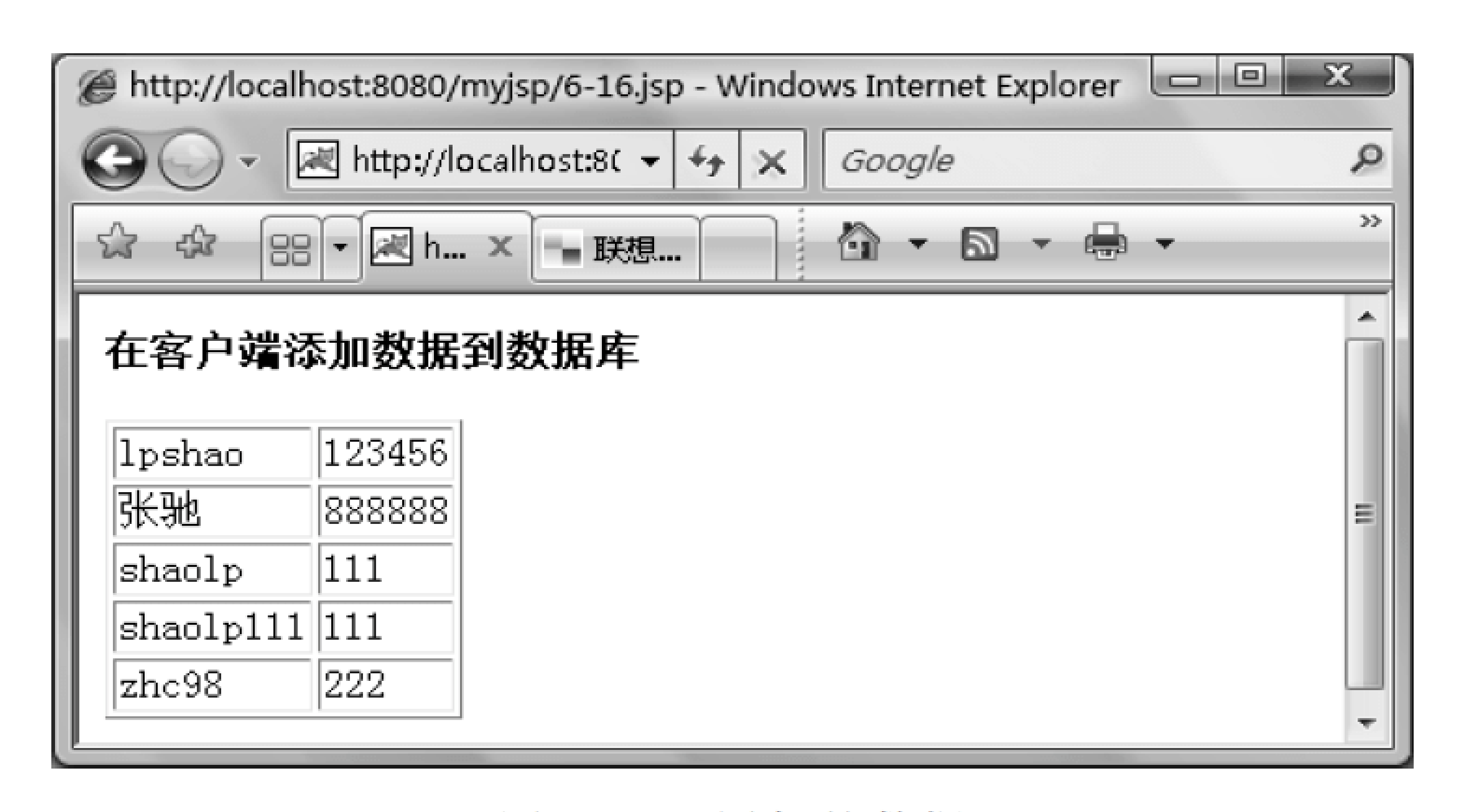


图 6.29　添加的数据

3. 知识点归纳

本例使用了包含文件，这样可以将常用的、共用的一些功能单独编写为一个独立的文件，可以使代码重用，并提高程序的可维护性。

6.5.3　在客户端输入查询条件并显示查询结果

能不能让用户在 Web 页面上输入查询数据的条件让数据库服务器检索特定的信息，然后将查询结果返回到用户的浏览器上呢？下面的例子就来解决这个问题。

例 6.17　按用户名查询用户口令的 JSP 文件(6-17.jsp)。

6-17.jsp 源代码如下：

```
<%@page contentType="text/html;charset=gb2312"%>
<form name="form1" method="post" action="6-17.jsp">
<p>用户名<input name="yhm" type="text" id="yhm" size="15"></p>
<p><input type="submit" name="Submit" value="执行查询">
<input type="reset" name=""value="重置"></font></p>
</form>
<%@include file="jdbc-jsp.jsp" %>
```

```
<%
String name=request.getParameter("yhm");
ps=c.prepareStatement("select * from admins where admname=?");
ps.setString(1,name);
ResultSet rs=ps.executeQuery();
%>
<HR>
按用户名查询的结果
<%@include file="table.jsp" %>
```

在浏览器中输入 http://localhost:8080/myjsp/6-17.jsp，将显示如图 6.30 所示页面，首先输入查询条件，然后按“执行查询”按钮，如果数据库中存在该用户名将显示其口令。



图 6.30　输入查询条件并显示查询结果的页面

6.5.4　一个简单的会员表维护系统

下面将通过几个 JSP 程序文件，构成一个 Web 应用系统，通过客户端对存放在服务器上 questionnaire 数据库中的会员表 admins 进行显示、更改、添加或删除的操作。

1. 会员表维护系统组成

会员表维护系统包含如下文件：保存存放数据的会员表 admins 的数据库文件 questionnaire、连接数据库的 JSP 文件 jdbc-jsp.jsp、会员表维护主界面的 JSP 文件 index.jsp、组成主界面“添加记录”部分的 HTML 文件 adduser.htm、添加记录到数据库的 JSP 文件 adduser.jsp 等 9 个文件。

2. 会员表维护主界面

例 6.18　会员表维护系统的所有功能都体现在其主界面(index.jsp)上，在该页面可以进行添加、更改、删除的操作，其具体功能的实现交由其他 JSP 文件处理，index.jsp 的界面如图 6.31 所示。

index.jsp 代码如下：

会员表维护主界面

添加记录

用户名称

用户密码

确认密码

提交 重置

当前所有会员		
lpshao	更改密码	删除会员
张驰	更改密码	删除会员
shaolp	更改密码	删除会员
shaolp111	更改密码	删除会员
zhc98	更改密码	删除会员

图 6.31　会员表维护主界面

```
<%@page contentType="text/html;charset=GB2312"%>
<%@include file="jdbc-jsp.jsp"%>
<div align="center"><font color="red">会员表维护主界面</font></div>
<jsp:include page="adduser.htm" flush="true"/>
<div align="center">
    <table bordercolor="#999999" cellspacing="0" cellpadding="0" width="50%" border="
1">
        <tr bgcolor="#999999">
            <td colspan="2"><div align="center">当前所有会员</div></td>
        </tr>
        <tr><td colspan="2">
<table cellspacing="0" cellpadding="0" width="100%" align="center" border="1"
bordercolor="#999999">
<%
String sql="select * from admins";
ResultSet rs=s.executeQuery(sql);
String adminname;
while(rs.next()) {
  adminname=rs.getString(1);
  out.print("<tr>");
  out.print("<td align=center>"+adminname+"</td>");
  out.print("<td align=center><a href=edituser.jsp?user="+adminname+">更改密码</a
></td>");
   if(!adminname.equals("admin"))
     {out.print("<td align=center><a href=deltuser.jsp?user="+adminname+">删除会员</
a></td>");}
   else
     out.print("<td align=center><font color=red>超级用户不能删除</font></td>");
     out.print("</tr>");
```

```
}
%>
</table></td></tr>
</table><hr></div>
```

3. 主界面中显示“添加记录”部分的 HTML 文件 adduser.htm

使用该文件是为了介绍 index 文件的代码，同时提供在 JSP 文件中引入 HTML 文件的方式。

adduser.htm 代码如下：

```
<hr><div align="center"><strong>添加记录</strong>
    <form name="form1" method="post" action=" adduser.jsp ">
        <table cellspacing="0" cellpadding="0" width="250" align="center" border="0">
            <tr><td>用户名称 <input class="smallinput" size="8" name="name"><br>
                    用户密码<input class="smallinput" type="password" size="8" name="
password"><br>
                    确认密码 <input class="smallinput" type="password" size="8" name="
confirm"><br>
                     </td>
            </tr>
            <tr><td><div align="center">
                    <input class="buttonface" type="submit" value="提交" name="submit">
                   <input class="buttonface" type="reset" value="重置" name="submit2">
            </div>
            </td>
        </tr>
      </table>
    </form>
</div><hr>
```

4. 添加记录到数据库的 JSP 文件 adduser.jsp

在主界面中添加的会员记录数据将提交给 adduser.jsp 处理，其代码如下：

```
<%@page contentType="text/html;charset=GB2312"%><%@include file="jdbc-jsp.jsp"%>
<%
String name=request.getParameter("name");
String password=request.getParameter("password");
String confirm=request.getParameter("confirm");
String sql="select * from admins where admname='"+name+"'";
ResultSet rs=s.executeQuery(sql);
out.print("<center>");
String id=null;
while (rs.next())
  id=rs.getString(1);
```

```
if(name.length()==0) {
  String errmsg="用户名字段不可为空白!";
  out.print("<font color=green size=5>错误信息<hr></font><font color=red>"+errmsg+
"</font><hr>");
}
else if(id!=null) {
  String errmsg="这个用户名已经有人在使用了,请换新的用户名!";
  out.print("<font color=green size=5>错误信息<hr></font><font color=red>"+errmsg+
"</font><hr>");
}
else if(password.length()==0) {
  String errmsg="密码不可为空!";
  out.print("<font color=green size=5>错误信息<hr></font><font color=red>"+errmsg+
"</font><hr>");
}
else if(!password.equals(confirm)) {
  String errmsg="请重新确认密码!";
  out.print("<font color=green size=5>错误信息<hr></font><font color=red>"+errmsg+
"</font><hr>");
}
else {
  String sql1="insert into admins(admname,admpwd) values ('"+name+"','"+password+"')";
  s.executeUpdate(sql1);
  response.sendRedirect("index.jsp");
}
//能刷新主界面
out.print("<input type=button value=回上一页 onclick=history.back();>");
%>
```

本程序有验证输入数据是否为空、是否确认,添加数据到数据库的两个功能。程序中使用了 response 对象的重新定址方法:response.sendRedirect("index.jsp")。

5. 处理主界面中"删除会员"操作的 JSP 文件 deltuser.jsp

deltuser.jsp 程序代码如下:

```
<%@page contentType="text/html;charset=GB2312"%>
<%@include file="jdbc-jsp.jsp"%>
<%
String deluser=request.getParameter("user");
String sql="delete from admins where admname='"+deluser+"'";
s.executeUpdate(sql);
response.sendRedirect("index.jsp");
out.print("<input type=button value=回上一页 onclick=history.back();>");
%>
```

程序中,"user"是在单击“删除用户”链接时传递过来的数据;执行删除 SQL 语句使用了更新数据的 executeUpdate 方法：s. executeUpdate(sql);。

6. 处理“更改密码”操作界面的 JSP 文件 edituser. jsp

edituser. jsp 程序代码如下：

```
<%@page contentType="text/html;charset=GB2312"%>
<%String name=request.getParameter("user");%>
<div align="center">
    <hr><center>更改会员密码
    <form name="form1" method="post" action="moduser.jsp">
        <table cellspacing="0" cellpadding="0" width="250" align="center" border="0">
            <tr><td>会员名称
                  <input class="smallInput" size="8" readonly name="name" value=
"<%=name%>"><br>
                会员密码<input type="password" size="8" name="password"><br>
                确认密码<input type="password" size="8" name="confirm"><br><br><br
></td>
            </tr>
            <tr><td><div align="center">
                <input type="submit" value="确认" name="Submit">
                <input type="reset" value="复位" name="Submit2"><br><br></div>
            </td>
        </tr>
      </table>
    </form>
</center></div>
```

edituser. jsp 程序提供一个更改会员密码的窗口页面,如图 6. 32 所示。其中会员名称是指定的。

图 6. 32　更改密码界面

7. 将更改的密码数据添加到数据库的 JSP 文件 moduser. jsp

moduser. jsp 程序代码如下：

```
<%@page contentType="text/html;charset=GB2312"%><%@include file="jdbc-jsp.jsp"%>
<%
String name=request.getParameter("name");
String password=request.getParameter("password");
String confirm=request.getParameter("confirm");
if(password.length()== 0) {
  String errmsg="密码不可为空!";
  out.print("<font color=green size=5>错误信息<hr></font><font color=red>"+errmsg+
"</font><hr>");
}
```

```
else if(!password.equals(confirm)) {
  String errmsg="请重新确认密码!";
  out.print("<font color=green size=5>错误信息<hr></font><font color=red>"+errmsg+
"</font><hr>");
}
else {
  String sql="update admins set admpwd='"+password+"'";
  s.executeUpdate(sql);
  response.sendRedirect("index.jsp");
}
out.print("<input type=button value=回上一页 onclick=history.back();>");
%>
```

这几个程序虽然简单,但包含了数据库处理的主要方法,参考其方式可以解决一般数据库处理的问题。在程序中 JSP 代码与 HTML 代码的混合使用方式也要好好体会。

6.5.5 连接 Access 数据库的不同方式

1. 连接带有用户名与密码的数据库

为了数据库的安全,可在 Access 数据库中设置用户名与密码,例如设置用户名为 lpshao、密码为 123456,数据源为 jsp,然后可使用下面的程序与数据源进行连接。

例 6.19 与 Access 数据库连接的程序(jdbc-access1.jsp)。

```
<%@page import="java.sql.*"%>
<%Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection("jdbc:odbc:jsp","lpshao","123456");
    %>
```

2. 直接连接数据库的方式

没有设置数据源,可以采用直接与数据库建立连接的方式,看下面的程序。

例 6.20 与 Access 数据库直接连接的程序(jdbc-access2.jsp)。

```
<%@page import="java.sql.*"%>
<%
try {Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); }
catch(ClassNotFoundException e1) {e1.printStackTrace(); }
String url="jdbc:odbc:driver={Microsoft Access Driver (*.mdb)};DBQ=C://WZKF// myjsp //
DBA \\数据库名.mdb";
Connection c=DriverManager.getConnection(url);
%>
```

6.6 思考与练习

6.6.1 思考题

6-1 什么是 SQL?

6-2　SQL 与 JSP 存在什么关系？

6-3　JSP 如何访问不同数据库中的数据？

6.6.2　上机练习

6-1　安装 Access 数据库。

6-2　在 Access 数据库中建立一个库存管理数据库文件 kcgl.mdb，并创建订单表 DD(ddh(订单号)、hh(货号)、pm(品名)、dw(订货单位)、sj(售价)、dgl(订购量)、dd(送货地点)、rq(送货日期，zt(订单处理状态)))、库存表 KC(货号，品名，库存量，仓库地点，单价)。

6-3　根据上面创建的数据库 kcgl，编写 SQL 语句查询下列问题：

(1) 查询所有订单。

(2) 10 天内要送货的货物的名称和送货地点、送货时间。

(3) 北京光明公司所订货物的名称和送货地点。

(4) 售价最高的货物。

(5) 订购量超过 2000 件的单位。

6-4　在计算机中使用 Window 操作系统的 ODBC 创建一个连接数据库文件 kcgl.mdb 的名称为 kcglDB 的数据源。

6-5　编写一个连接库存管理数据库 kcgl 的 jdbc-kcgl.jsp 文件。

6-6　使用 JSP 技术编写一个订货单处理系统，包括以下功能：

(1) 输入订单功能页面，通过一个静态表单页面输入订货信息，然后通过一个 JSP 程序将输入订货信息存放到数据库 kcgl 的订单表 DD 中。

(2) 使用 JSP 技术编写一个显示所有订货单的页面。

(3) 使用 JSP 技术编写一个按订单号查询订货单的页面，用户可以在页面上输入订单号，提交后显示该订单号的货物名称、定购量和送货地点。

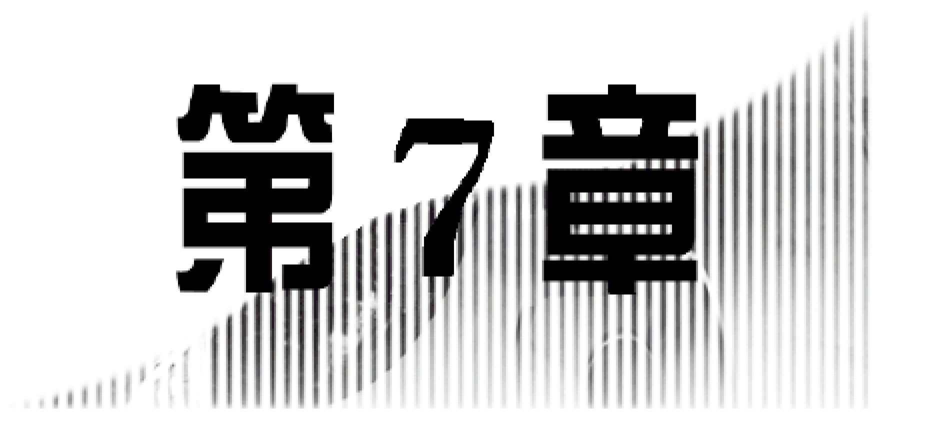
(4) 使用 JSP 技术编写一个按订单日期查询显示相应订货单的页面，可以选择“订单日期”的范围，然后按日期范围查询并显示订单号、货物的名称、定购量和送货地点。

6-7　使用 JSP 技术编写一个输入与显示新闻的处理系统，包括以下功能：

(1) 一个显示新闻标题、发布新闻时间、新闻主要内容的页面。

(2) 一个输入新闻标题、发布新闻时间、新闻主要内容的页面(提交到显示页面并刷新页面)。

(3) 在新闻主要内容中包含一个超链接到全部新闻页面(HTML 文件)。



第7章

可扩展标记语言——XML

可扩展标记语言(eXtensible Markup Language,XML)正被迅速的广泛运用,它已作为与平台、语言和协议无关的格式描述和交换数据的广泛应用标准。XML 和它的辅助规范可用于描述数据的文档表现,描述 XML 文档类型的限制,描述 XML 文档和资源之间的链接,描述 XML 文档的自动转换和格式化。

本章主要解决以下问题:

- 什么是 XML;
- 什么是 XML 的命名空间;
- XML 文档由什么构成;
- 什么是 XML 模式定义;
- 如何验证 XML 文档的有效性;
- 如何通过 JSP 获取 XML 中的数据;
- 如何使用 JDOM 处理 XML 数据。

7.1 XML 概述

XML 是英文 eXtensible Markup Language 的缩写,中文含义为可扩展标记语言。它是万维网协会(W3C)推出的新一代数据交换的标准。目前推荐遵循的是 W3C 组织于 2000 年 10 月 6 日发布的 XML 1.0 版本(可到 www.w3.org/TR/2000/REC-XML-2001006 了解相关内容)。和 HTML 一样,XML 同样来源于通用标记语言(SGML),但 XML 与 HTML 有很大的不同,HTML 的主要目标是描述文本的结构与样式,其主要任务是显示文本与控制文本,而 XML 是一种协议语言,主要目标是描述数据和数据结构,其主要任务是定义 Web 网页上的文档元素和商业文档,以及用于复杂结构数据的表示与传输。

XML 可以看成为一种简单的数据存储语言,它使用一系列简单的标记来描述数据,这些标记可以用方便的方式建立,通过 XML 文档可以存储数据。虽然使用 XML 数据占用的空间比二进制数据占用更多的空间,但 XML 文档极其简单,易于掌握和使用。XML 与 Access,Oracle 和 SQL Server 等数据库不同,数据库提供了更强有力的数据存储和分

析能力，例如：数据索引、排序、查找、相关一致性等，XML 仅仅用来定义数据。事实上 XML 与其他数据表现形式最大的不同就是：它极其简单，正是这个优点使 XML 与众不同。

本节主要介绍 XML 文档的结构及基本语法。

7.1.1 XML 文档的元素与标记

XML 可以看成为一种描述数据的标准，由一组定义语义标记的规则组成，这些语义标记将文档划分为多个部分，并且标记出文档的不同部分。如果一个文档满足 XML 1.0 标准“格式正确(Well-Formed)”的定义，则该文档是一个 XML 文档，文档带有.xml 文件扩展名。

先来看一个具体的 XML 文档。

1. 创建 XML 文档

例 7.1 一个描述联系人数据的 XML 文档(7-1.xml)。

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<!--xml 文档结构及语法举例-->
<booklist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
noNamespaceSchemaLocation="7-1.xsd">
    <book sales="Y">
        <编号>F8915</编号>
        <title>网站编程技术实用教程</title>
        <authorlist>
            <author>邵丽萍</author>
            <author>张后扬</author>
        </authorlist>
        <price>580</price>
    </book>
    <book sales="N">
        <编号>F8916</编号>
        <title>网站架设彻底研究</title>
        <authorlist>
            <author>杜晖</author>
        </authorlist>
        <price>550</price>
    </book>
</booklist>
```

看上去 XML 文档的书写格式与 HTML 非常相似，但这里的标记代表的不再是显示格式，而是对于数据的语意解释。

7-1.xml 文件清晰描述了两本书的编号、书名、作者与书价等信息。

2. XML 元素

由 7-1.xml 文件可以看到 XML 文档是由一些元素构成的。XML 文档的元素有什么特征呢？

1）XML 元素的结构

一个 XML 文档的元素包含一个开始标记，一个结束标记与元素内容。元素内容可以是字符数据或其他元素（子元素）。

元素的语法格式：

```
<开始标记>元素内容</结束标记>
```

例如，例 7.1 中 price 是一个元素的元素名，其开始标记为<price>，结束标记为</price>，元素内容为字符数据 580。

2）空元素

没有元素内容的元素称为空元素，例如<标记名></标记名>，空元素可以写为<标记名/>。

3）根元素

从文档头开始，一直到文档尾，包含文档中所有数据的元素称为根元素，也称为文档标记，例 7.1 中 booklist 就是一个根元素。

4）简单元素与复杂元素

复杂元素中可以包含子元素。例如，例 7.1 中 booklist 是一个复杂元素的元素名，该元素包含 2 个子元素 book，因此是复杂元素。同样，book、authorlist 也是复杂元素。

简单元素的元素内容只能是字符数据，例如，例 7.1 中的 authorlist 元素中包含的子元素 author 为简单元素，它的开始标记为<author>，结束标记为</author>，元素内容为字符数据“邵丽萍”。

3. XML 标记

1）开始标记格式

```
<标记名 [属性名="属性值"] * >
```

例如，例 7.1 中的<book sales="N">就是一个带有属性的开始标记。

2）结束标记格式

```
</标记名>
```

例如，例 7.1 中的</book>为结束标记。

3）标记的注意事项

- 标记区分大小写；
- 标记严格配对；
- 标记以字母、文字、下划线、冒号开头，后跟字母、数字、句号、冒号、下划线或连字符，不能有空格，不能以 xml 开始；

- 标记中可以包含任意多个属性,属性间用空格分隔,属性以属性名="属性值"的形式出现,属性值必须用单引号或双引号括起来。

4. 字符数据的实体引用

开始标记和结束标记之间的文字称为字符数据。字符数据如果是一些特殊符号时,需要采用它的实体引用来代替,特殊符号的实体引用如表 7.1 所示。

表 7.1 实体引用

特殊符号	实体引用	特殊符号	实体引用
>	>	“	"
<	<	‘	'
&	&		

例如,<示例><姓名 >John</姓名 ></示例>。其中,<用来代替“<”,> 用来代替“>”。

5. XML 注释语法格式

```
<!--XML 注释-->
```

XML 的注释方式与 HTML 相同。

6. CDATA 标记格式

```
<![CDATA[文本内容]]>
```

在标记 CDATA 下,所有的标记、实体引用都被忽略,而被 XML 处理程序当作字符数据看待。例如:

```
<示例><![CDATA[<姓名>John</姓名>]]></示例>
```

其中“<姓名>John</姓名>”被视为字符数据。

7. XML 与 HTML 的不同点

XML 与 HTML 的不同体现在以下方面:

(1) HTML 侧重于如何显示文本;XML 侧重于如何结构化地描述数据。

(2) HTML 不要求标记的嵌套、配对,不要求标记之间具有一定的顺序;XML 严格要求嵌套、配对,并遵循树形结构。

(3) HTML 文档中不区分大小写;XML 文档中区分大小写。

(4) HTML 文档中属性值的写法没有严格规定;XML 文档中属性值必须写在引号中。

(5) HTML 文档中的标记是定义好的,一般为英文字母或英文单词;XML 文档中的标记码可以随意定义,可以是英文或中文等符号。

7.1.2 XML 命名空间

XML 的命名空间是 W3C 推出的一个标准，它提供了一种辨识标记信息，避免标记名冲突的方法。

1. 标记名冲突的例子

由于 XML 中的元素名是预定义的，当两个不同的文档或同一文档使用相同的元素名时，就会发生命名冲突。

例 7.2 一个带有命名冲突的 XML 文档(7-2. xml)。

```
<?xml version="1.0"encoding="gb2312"?>
<booklist>
    <book>
        <编号>F8915</编号>
        <title>网站编程技术实用教程</title>
        <author>
            <name>张后扬</name>
            <title>教授</title>
        </author>
        <price>580</price>
    </book>
</booklist>
```

在 7-2. xml 文档中有两个相同的标记名“title”。

例 7.3 一个简单的 XML 文档(7-3. xml)。

```
<?xml version="1.0"encoding="gb2312"?>
<authorlist>
    <author>
        <name>张后扬</name>
        <title>教授</title>
    </author>
</authorlist>
```

例 7.4 一个简单的 XML 文档(7-4. xml)。

```
<?xml version="1.0"encoding="gb2312"?>
<authorlist>
    < author>
        <name>杜晖</name>
        <title>博士</title>
    </author>
</authorlist>
```

如果 7-3. xml 与 7-4. xml 这两个 XML 文档一起使用，由于两个文档都包含带有不

同内容和定义的<authorlist>元素,将会发生命名冲突。

如何避免命名冲突问题呢?

2. 使用前缀来避免命名冲突

在标记名前添加一个前缀可以区别两个相同的标记名。带有前缀的标记名称为完整的标记名。

完整的标记名的语法格式如下:

```
<前缀:标记名>
```

例如,<al:title >,<bl:title >即可区别两个相同的标记名。al 与 bl 为前缀。

3. XML 的命名空间标识符

为了彻底解决命名冲突的问题,W3C 推出了一个标准,即 XML 的命名空间。使用命名空间可以明确标识出 XML 文档中元素、属性以及其他标记,可以避免名称之间所带来的冲突问题。

在 XML 文档中,一般都使用统一资源标识符(Universal Resource Identifier,URI)作为命名空间标识符,URI 是用字符串唯一标识信息资源(网页、要下载的文件等)的工业标准,URI 有两种常规类型:统一资源定位器(URL)和统一资源名称(Universal Resource Name,URN)。这两种类型的 URI 都可以用作命名空间标识符。URL 主要用于唯一标识网页的空间地址。URN 主要用于唯一标识全球范围内由专门机构负责的稳定的信息资源,通常给出资源名称而不提供资源位置。

命名空间标识符最重要的属性是它的唯一性。使用者可以通过向 Internet 命名机构注册域名来保证 URL 与 URN 的唯一性,并确保域名后面使用的所有字符串都保持唯一。

4. XML 命名空间的声明

由于一般的 URI 都比较长,放在元素和属性名称前难于书写,因此,常使用一个简短的字符来代替 URI,并称之为命名空间的前缀。

声明命名空间有三种形式:

1) 默认的命名空间声明方式

如果 XML 文档中所有元素属于同一命名空间,则只需在根元素声明命名空间,语法格式为:

```
<命名空间前缀:根元素标记名 xmlns:命名空间前缀="URI">
```

例 7.5 一个带有默认声明命名空间的 XML 文档(7-5.xml)。

```
<?xml version="1.0"encoding="gb2312"?>
<!--默认声明命名空间-->
<bl:booklist xmlns:bl="http://www.company1.com">
    <book>
```

```
        <编号>F8915</编号>
        <title>网站编程技术实用教程</title>
        <price>580</price>
    </book>
</bl:booklist>
```

2）显式声明命名空间的方式

如果 XML 文档中不同元素属于不同命名空间，可在根元素中显式声明不同的命名空间，语法格式为：

```
<根元素标记名 xmlns:命名空间前缀 1="URI1" xmlns: 命名空间前缀 2="URI2" …>
```

例 7.6 一个带有显式声明命名空间的 XML 文档(7-6.xml)。

```
<?xml version="1.0"encoding="gb2312"?>
<!--显式声明命名空间-->
<booklist xmlns:bl="http://www.company1.com" xmlns:al="http://www.company2.com">
    <bl:book>
        <bl:编号>F8915</bl:编号>
        <bl:title>网站编程技术实用教程</bl:title>
        <al:author>
            <al:name>张后扬</al:name>
            <al:title>教授</al:title>
        </al:author>
        <bl:price>580</bl:price>
    </bl:book>
</booklist>
```

3）隐式声明命名空间的方式

如果 XML 文档中不同元素属于不同命名空间，可在不同元素中隐式声明不同命名空间，语法格式为：

```
<元素标记名 xmlns="URI1">
<元素标记名 xmlns="URI2">
```

例 7.7 一个带有隐式声明命名空间的 XML 文档(7-7.xml)。

```
<?xml version="1.0"encoding="gb2312"?>
<!--隐式声明命名空间-->
<booklist>
    <book>
        <编号>F8915</编号>
        <title xmlns="http://www.company1.com">网站编程技术实用教程</title>
        <author>
            <name>张后扬</name>
            <title xmlns="http://www.company2.com">教授</title>
        </author>
```

```
        <price>580</price>
    </book>
</booklist>
```

7.1.3 XML 文档结构

1. XML 文档的组成

XML 文档包含 4 个部分：XML 文档声明、根元素、元素与属性。

XML 文档声明一般放在 XML 文档的第一行，用来说明 XML 文档的版本号，文档编码等。例如：

```
<?xml version="#"encoding="#"?>
```

XML 声明由“<?”开始，到“?>”结束。在“<?”后面紧跟着处理指令的名称“xml”，它用来告诉浏览器下面处理的是一个 XML 文档。

在 XML 文档声明中，version 属性指明 XML 的版本号；encoding 属性指明 XML 文档使用的编码类型。

2. XML 文档的树型结构

每一个 XML 文档都具有树型结构，7-1.xml 对应的树型结构如图 7.1 所示。

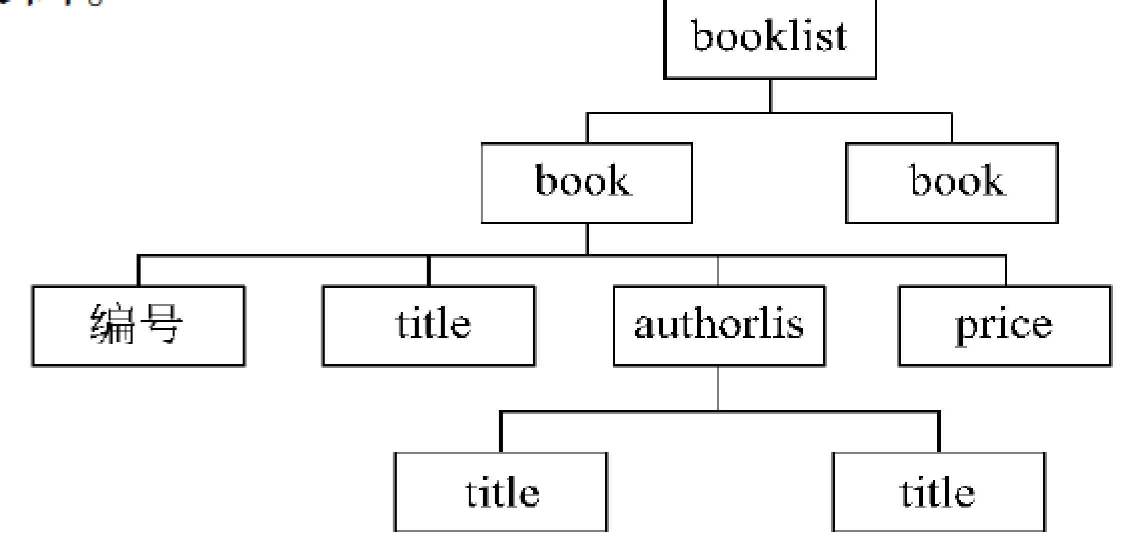


图 7.1　XML 文档的树型结构

7.1.4 XML 模式定义

1. 什么是 XML Schema

XML 模式定义（XML Schema Definition，XSD）或称为 XML Schema，是取代文档类型定义（Document Type Definition，DTD）针对 XML 的新一代验证机制，它用来检查 XML 文档的结构、标记和属性是否符合规范要求。

2. XML Schema 的作用

XML 文档只说明数据的结构，数据的具体描述、数据是否正确及数据类型都没有说明。这些问题由 XML Schema 解决，XML Schema 的主要任务就是定义和描述 XML 文档的结构和内容模式，即定义 XML 文档中元素之间的关系、元素和属性的数据类型、XML 文档的结构以及文档内容的约束条件。

Schema 是一种保证 XML 文档格式正确的有效方法，通过比较 XML 文档和 Schema 文件可以确认 XML 文档是否符合规范，元素和标记使用是否正确。

XML 文件可以为应用程序提供一个数据交换的格式，而 Schema 能让 XML 文件成为数据交换的标准，不同的使用者只需定义好标准的 Schema 文件，不同的人都能够依照 Schema 文档建立 XML 文件，并且进行验证，如此就可以轻易地建立标准和交换数据，满

足网络共享和数据交互。

3. Schema 文档

XML Schema 文档的扩展名为.xsd。XML Schema 文档自身也是 XML 文档，使用 XML 语法。但其文档是用一套预先规定的 Schema 元素和属性创建的，这些元素和属性定义了文档的结构和内容模式。同时预先制定了一套规则确定每个 Schema 元素或者属性的合法用途。如果违反这些规则解析器就会拒绝解析 Schema 以及任何同它相联系的 XML 文档。

一个 XML Schema 文档主要包含：元素的定义规则，元素间关系的定义规则，元素可使用的属性，可使用的实体或符号规则等。

例 7.8 7-1.xml 文档的 Schema 文档(7-1.xsd)。

```
<?xml version="1.0" encoding="gb2312"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/
XMLSchema">
    <xs:element name="booklist">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="book" type="bookType" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="bookType">
        <xs:sequence>
            <xs:element name="编号" type="xs:string"/>
            <xs:element name="title" type="xs:string"/>
            <xs:element name= "authorlist" type= "authorlistType"/>
            <xs:element name="price" type="xs:decimal"/>
        </xs:sequence>
        <xs:attribute name="sales" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="N"/>
                    <xs:enumeration value="Y"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="authorlistType">
        <xs:sequence>
            <xs:element name="author" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

4. XML Schema 文档的组成

1）XML Schema 文档声明

从 7-1. xsd 的第 1 行可以看出，XML Schema 文档实际上是一个 XML 文档，只是使用的文档扩展名不同。

2）XML Schema 命名空间

从 7-1. xsd 的第 2 行可以看出，XML Schema 默认使用 W3C 命名空间：xmlns:xs="http://www.w3.org/2001/XMLSchema";。

elementFormDefault 属性值为"qualified"表示元素需要使用前缀；属性值为"unqualified"表示元素不需要使用前缀。

3）XML Schema 的主要元素

预先规定的 Schema 主要元素如表 7.2 所示。

表 7.2 XML Schema 的主要元素

元　素	说　明
schema	XML Schema 文档的根元素
element	声明 XML 元素的元素
attribute	声明 XML 元素属性的元素
simpleType	声明 XML 元素为简单数据类型的元素
complexType	声明 XML 元素为复杂数据类型的元素
sequence	声明 XML 子元素需要按顺序出现的"标记名"元素
all	声明 XML 子元素可以以任意顺序出现的"标记名"元素

4）element 元素的语法格式

```
<element name="#", type="#", minOccurs="#", maxOccurs="#"/>
```

- name 属性说明 XML 元素的名称；
- type 属性说明 XML 元素内容的数据类型；
- minOccurs 属性说明元素最少出现的次数；
- maxOccurs 属性说明元素最大出现的次数，其值为 unbounded 表示无限次数。

5）attribute 元素的语法格式

```
<attribute name="#", type="#", use="#", default="#", fixed="#"/>
```

- name 属性说明 XML 元素的属性名称；
- type 属性说明 XML 元素属性内容的数据类型；
- use 属性说明 XML 元素属性的使用方式，值 optional 表示属性可有可无，值 required 表示属性需要出现一次；
- default 属性说明属性的默认值；

- fixed 属性说明属性的固定值。

6）simpleType 简单元素的语法格式

```
<xs:simpleType name="#">
    <xs:restriction base="xs:#">
        <xs:facets_element value="#"/>
        ⋮
    </xs:restriction>
</xs:simpleType>
```

- name 属性说明数据类型的名称。
- restriction 子元素和 base 属性说明使用的基本数据类型，XML Schema 内置的基本数据类型如表 7.3 所示。

表 7.3　XML Schema 内置的数据类型

数据类型	说　　明	数据类型	说　　明
string	字符串数据	integer	整数
decimal	任意精度和位数的十进制数	float	标准的 32 位浮点数
date	日期数据，格式为 YYYY-MM-DD	dateTime	历法日期和时间
boolean	二元类型的 True 或者 False 数据	time	24 小时格式的时间可根据时区调节

- facets_element 子元素描述数据类型的细节限制，它具有多种形式，如表 7.4 所示。

表 7.4　facets_element 的不同形式

facets_element 子元素	说　　明
pattern	声明元素内容为正则表达式的子元素
minLength	声明元素内容最小长度的子元素
maxLength	声明元素内容最大长度的子元素
enumeration	声明在有限的元素内容中选一的子元素

例如，要定义字符串数据的长度最少为 1、最大为 6，可使用如下语句：

```
<xs:simpleType name="productCode">
  <xs:restriction base="xs:string">
      <xs: minLength value="1"/>
      <xs: maxLength value="6"/>
  </xs:restriction>
</xs:simpleType>
```

要定义“电话号码”字符串数据为 0～9，样式为 010-51688529，可使用如下语句：

```
<xs:simpleType name="phoneNumber">
```

```
    <xs:restriction base="xs:string">
        <xs: pattern value="\d{3}-\d{8}"/>
    </xs:restriction>
</xs:simpleType>
```

其中,使用了正则表达式,即由普通字符(例如字符 a～z 及-)以及特殊字符(称为元字符,例如\d)组成的文字模式(见表 7.5)。该模式描述在查找文字主体时待匹配的一个或多个字符串。正则表达式作为一个模板,将某个字符模式与所搜索的字符串进行匹配。

表 7.5 常用的特殊字符

特殊字符	说 明
\d	匹配一个数字字符。等价于 [0-9]
{*n*}	*n* 是一个非负整数。确定匹配 *n* 次
\w	匹配包括下划线的任何单词字符。等价于[A-Za-z0-9_]
[*xyz*]	匹配所包含的任意一个字符。例如,[abc]可以匹配 plain 中的 a
(pattern)	匹配 pattern 并获取这一匹配
*	匹配前面的子表达式零次或多次
+	匹配前面的子表达式一次或多次。例如,'zo+'能匹配"zo"以及"zoo",但不能匹配"z"

要定义“电子邮箱地址”字符串数据,样式为 zhc111@sina.com,可使用如下语句:

```
<xs:simpleType name="SEmail">
    <xs:restriction base="xs:string">
        <xs:maxLength value="30"/>
        <xs:pattern value="\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"/>
    </xs:restriction>
</xs:simpleType>
```

7) complexType 复杂元素的语法格式

如果声明的 XML 元素拥有属性和子元素,则使用 complexType 元素声明其属性及子元素。

```
<xs:complexType name="#">
   <xs:标记名元素>
       <xs:element ….>
         ⋮
   </xs:标记名元素>
   <xs:attribtue ….>
</xs:complexType>
```

name 属性说明数据类型的名称。

5. 使用 XML Schema 文档检验 XML 文档有效性的方式

在 XML 文档中指定验证使用的 XML Schema 文档,例如

```
<booklist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xsi:
noNamespaceSchemaLocation="7-1.xsd">
```

其中，xsi 为 XML 文档默认命名空间 http://www.w3.org/2001/XMLSchema-instance 的前缀。

xsi：noNamespaceSchemaLocation 属性指定不使用命名空间方式指定 XML Schema 文档的路径。这里表示 XML Schema 文档要存放在与验证的 XML 文档的同一路径下，例如，作者使用的是 myjsp 目录。

6. 编写 XML Schema 文档的工具

编写 XML Schema 文档可以使用一些辅助工具，XML Spy、XML Validator 和 Cape Clear Studio 都是具有 XML Schema 创建功能的全方位 XML 综合开发环境。

7.1.5 使用 FrontPage 2003 验证 XML 文档的格式

使用 FrontPage 2003 可以编写 XML 文档并验证其有效性。

下面介绍使用 FrontPage 2003 创建与验证 7-8.xml 文档的具体步骤。

1. 输入 XML 文档代码

启动 FrontPage 2003，在“代码”视图下输入代码，如图 7.2 所示。

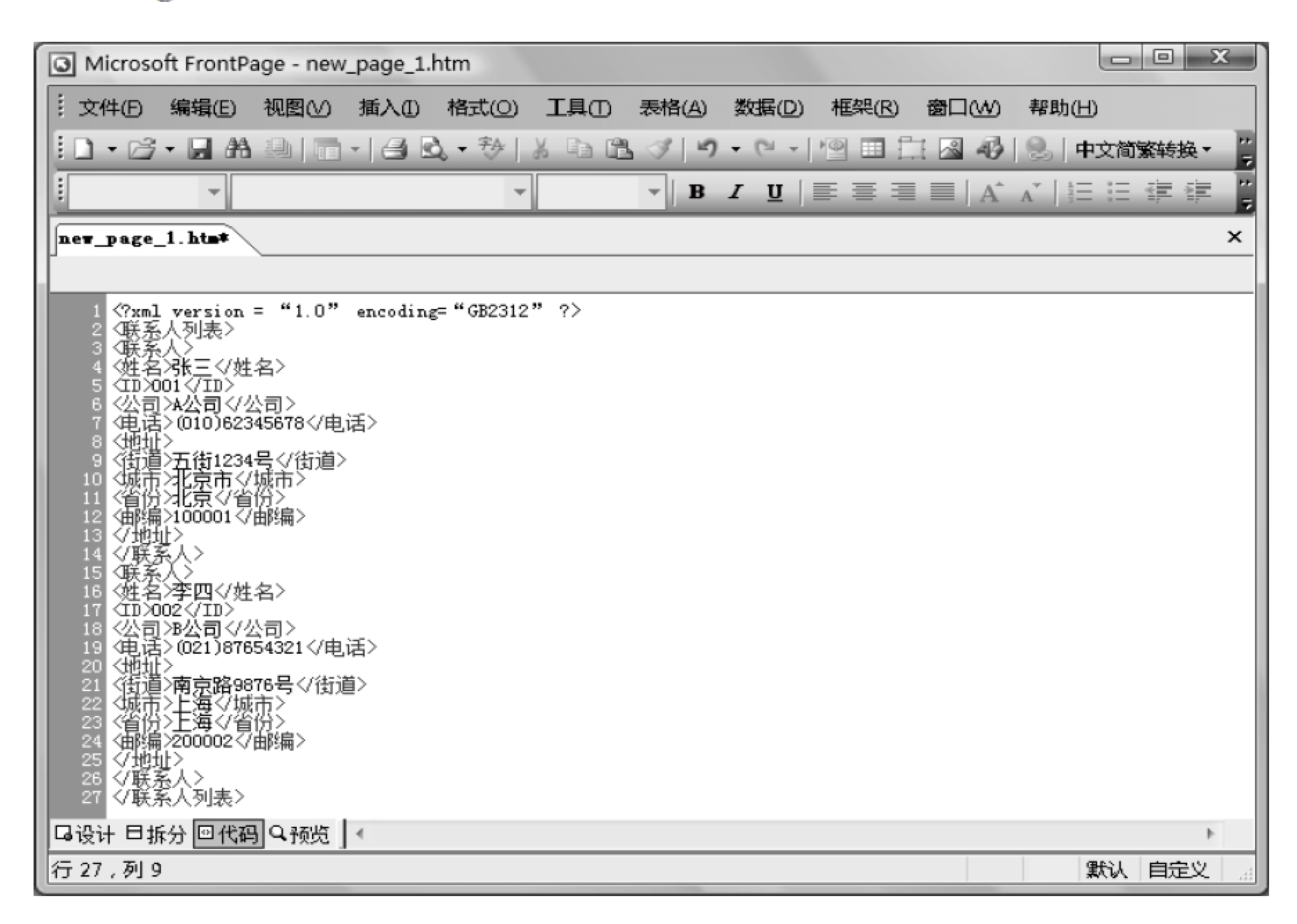


图 7.2 在 FrontPage 2003 中输入 XML 文档代码

2. 保存 XML 文档为 7-8.xml

在 FrontPage 2003 工具栏单击“保存”按钮，打开“另存为”对话框，在其中输入文件名 7-8，设置保存类型为“XML 文件”，并将文件保存在“C:\wzkf\myjsp”，如图 7.3 所

示。单击“保存”按钮。



图 7.3　在 FrontPage 2003 中保存 XML 文档

3. 验证 7-8.xml 文档

(1) 保存文件后，在 FrontPage 2003 菜单栏单击“视图”→“工具栏”→“XML 视图”菜单，打开“XML 视图”对话框，如图 7.4 所示。

图 7.4　“XML 视图”对话框

(2) 单击“重新设置 XML 格式”按钮，可以看到，XML 代码的格式被重新进行了调整。

(3) 单击“验证标准格式的 XML”按钮，将出现如图 7.5 所示的“文档的格式及嵌套均正确”的对话框，表明所编写的 XML 文档满足 XML 1.0 标准“良好格式”的定义，是一个具有良好格式的 XML 有效文档。否则，会出现出错信息，可根据提示修改文档。

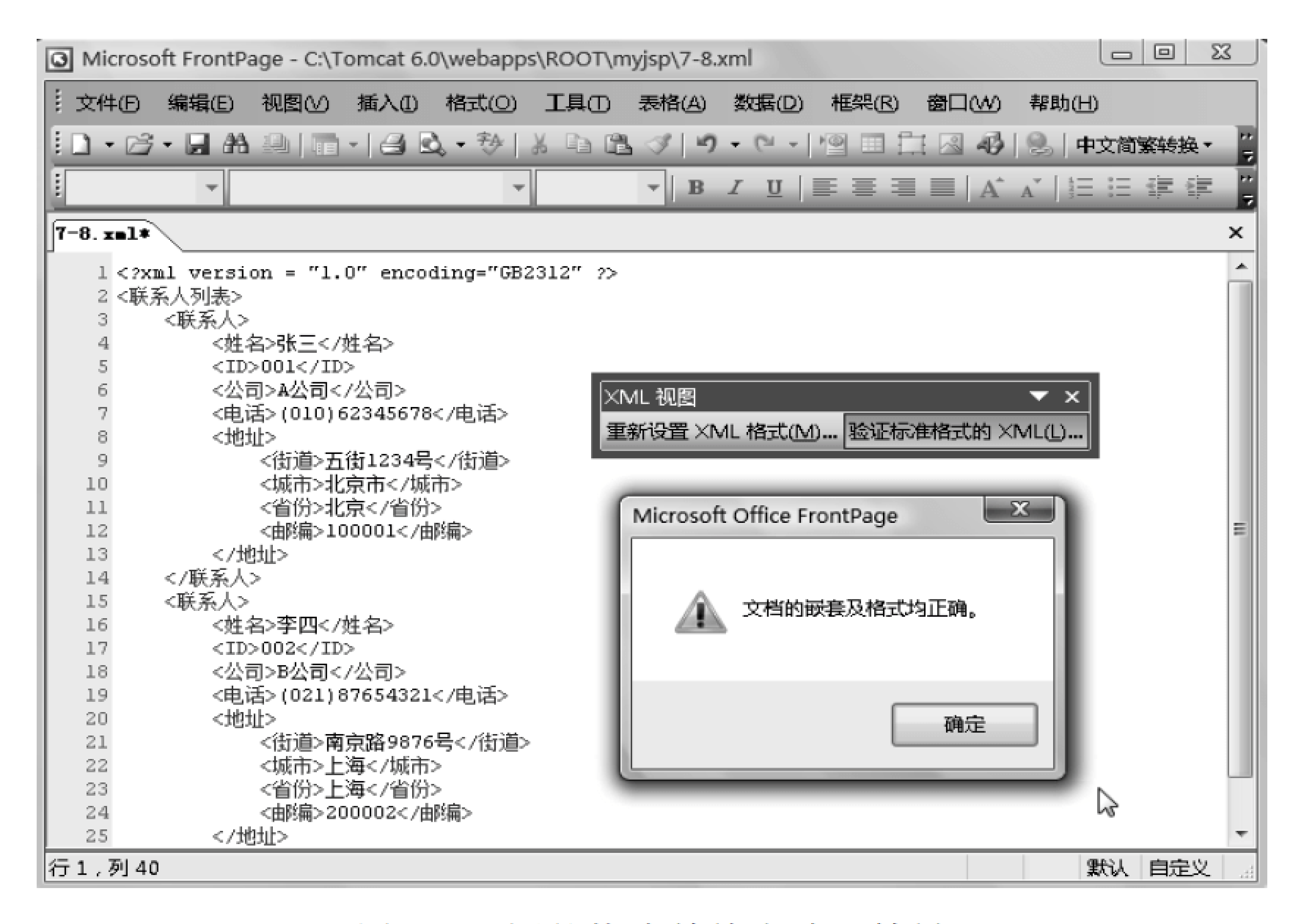


图 7.5　调整格式并给出验证结果

7.2 在浏览器中格式化显示 XML 文档中的数据

要在 IE 浏览器中格式化显示 XML 文档中的数据，可以使用 CSS 样式表文件或 XSLT 模式文件来实现。

本节的内容主要介绍在浏览器中格式化显示 XML 文档数据的方式。

7.2.1 使用 CSS 显示 XML 文档中的数据

1. 在 XML 文件中添加引用 CSS 样式表的语句

例 7.9 添加引用 CSS 样式表语句的 XML 文件(7-9.xml)。

```
<?xml version="1.0" encoding="gb2312" ?>
<?xml:stylesheet type="text/css" href="7-9.css"?>
<lz>
<reason>你想学 XML 吗?</reason>
<question>是不是很难学呀?</question>
<think>我是不是要学呢?</think>
<request>如果你要学习,好好看课件吧!</request>
</lz>
```

将 7-9.xml 文件存放在 myjsp 文件夹中。

2. 创建显示 XML 数据的 CSS 文件

例 7.10 显示 XML 数据的 CSS 样式表文件(7-9.css)。

```
lz{background-color: #00ffff; width:80%;}

reason {
font-family:"宋体"; font-size: 22pt;display: block;margin-bottom:10pt;margin-left:20;
}

question{font-family:"宋体";font-size:9pt;color:red; }
think{
font-family:"宋体";font-size: 9pt;block;Display:10pt;color:green;
}

request{
font-family:"宋体";font-size :9pt; Display: block; color: blue;margin-left: 20pt;
}
```

将 7-9.css 文件存放在 myjsp 文件夹中。在浏览器中输入 http://localhost:8080/myjsp/7-9.xml，可看到如图 7.6 所示页面。



图 7.6　使用 CSS 显示的 XML 数据

7.2.2　使用 XSL 显示 XML 文档中的数据

1. 在 XML 文件中添加引用 XSL 样式文件的语句

例 7.11　添加引用 XSL 样式文件语句的 XML 文件(lxr.xml)。

```
<?xml version="1.0" encoding="GB2312" ?>
<?xml-stylesheet type="text/xsl" href="lxr.xsl" ?>
<联系人列表>
<联系人>
<姓名>张三</姓名>
<ID>001</ID>
<公司>A公司</公司>
<电话>(010)62345678</电话>
<地址>
<街道>五街 1234 号</街道>
<城市>北京市</城市>
<省份>北京</省份>
<邮编>100001</邮编>
</地址>
</联系人>
<联系人>
<姓名>李四</姓名>
<ID>002</ID>
<公司>B公司</公司>
<电话>(021)87654321</电话>
<地址>
<街道>南京路 9876 号</街道>
<城市>上海</城市>
<省份>上海</省份>
<邮编>200002</邮编>
</地址>
</联系人>
```

```
</联系人列表>
```

将 lxr. xml 文件存放在 myjsp 文件夹中。

2. 创建显示 XML 数据的 XSL 文件

例 7.12 显示 XML 数据的 XSL 样式文件(lxr. xsl)。

```
<?xml version="1.0" encoding="GB2312"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" >
<!--根模板-->
<xsl:template match="/">
<HTML>
<HEAD>
<TITLE>F公司的客户联系信息</TITLE>
</HEAD>
<BODY>
<xsl:apply-templates select="联系人列表"/>
</BODY>
</HTML>
</xsl:template>
<!--联系人列表模板-->
<xsl:template match="联系人列表">
<xsl:for-each select="联系人">
<UL>
<LI><xsl:value-of select="姓名"/></LI>
<UL>
<LI>用户 ID:<xsl:value-of select="ID"/></LI>
<LI>公司:<xsl:value-of select="公司"/></LI>
<LI>电话:<xsl:value-of select="电话"/></LI>
<LI>街道:<xsl:value-of select="地址/街道"/></LI>
<LI>城市:<xsl:value-of select="地址/城市"/></LI>
<LI>省份:<xsl:value-of select="地址/省份"/></LI>
<LI>邮编:<xsl:value-of select="地址/邮编"/></LI>
</UL>
</UL>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

将 lxr. xsl 文件存放在 myjsp 文件夹中。在浏览器中输入 http://localhost:8080/myjsp/lxr. xml,可看到如图 7.7 所示页面。

如果将 XML 文件比作数据原料的话,那么 XSL 就好比"筛子"与"模型",XSL 可以筛选自己需要的数据原料,XSL 可以设计好表现数据的模型,并用 XML 中筛出的数据来填充模型,按照 XSL 定义的模型运行 XML 程序即可看到按模型定义的样式筛选出的

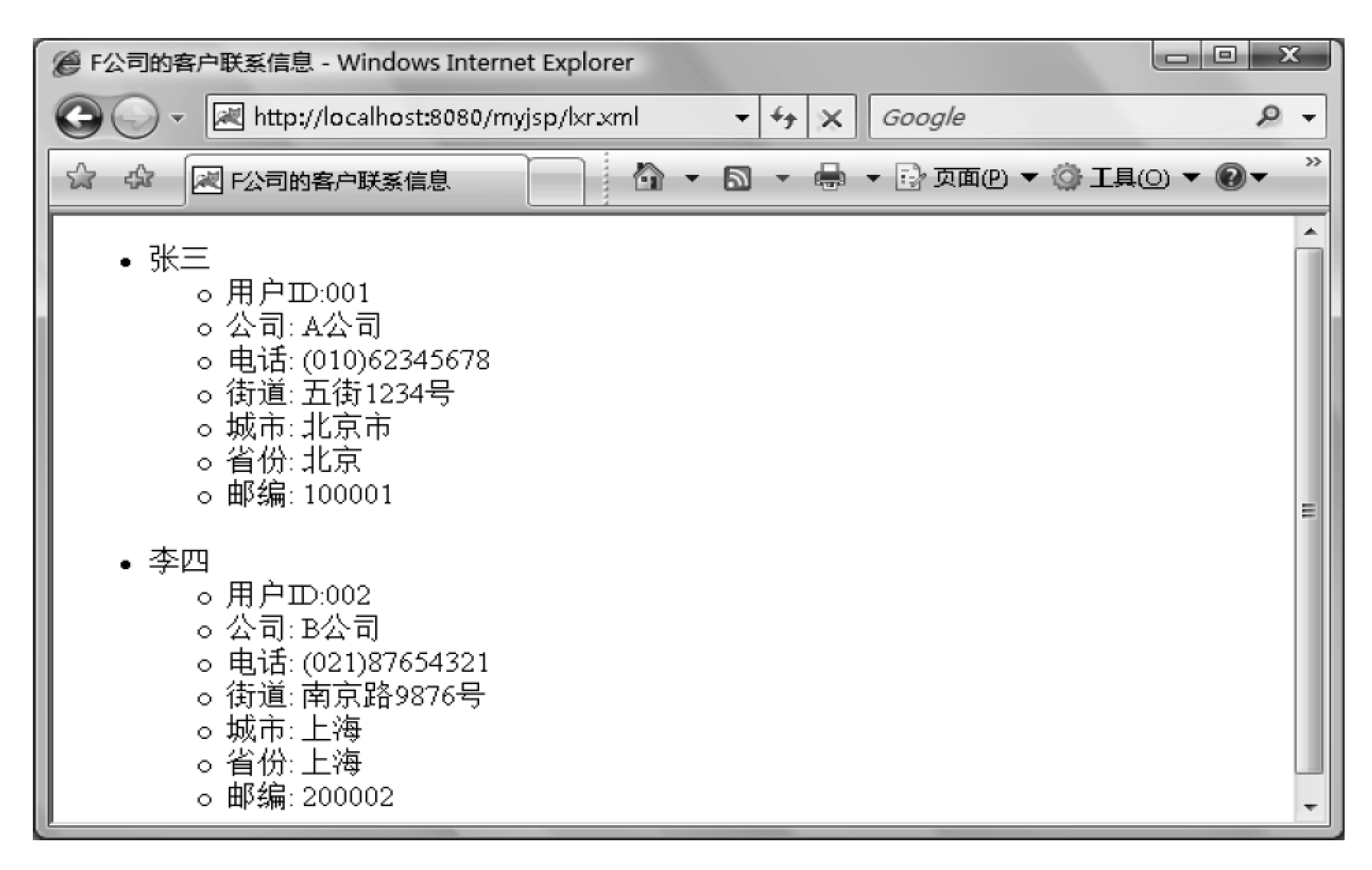


图 7.7　使用 XSL 显示的 XML 数据

XML 中的数据,其数据是通过 HTML 页面显示的。

3. XSL 文档的框架

XSL 文档由 3 部分构成:XML 文档声明、XSL 文档声明与 XSL 模板。

1) XML 文档声明

XSL 文档自身也是 XML 文档,所以,与 XML 文件一样,在 XSL 文档第 1 行为 XML 文档声明。

2) XSL 文档声明

XSL 文档中第 2 行和最末行通过 xsl:stylesheet 元素对声明这是一个 XSL 样式表文档。并在第 2 行开始标记名后,用 xmlns:xsl="URL"声明了命名空间,URL 为 http://www.w3.org/TR/WD-xsl。

3) XSL 模板

其他行可以用来定义根模板与其他模板,在例 7.12 中定义了一个根模板与一个联系人列表模板。XSL 文档中必须有根模板。

XSL 文档是由多个模板组成的,这些模板组合成一个完整的 XSL 模型。好比船与集装箱,先将货物装在不同的集装箱中,再将集装箱堆放在船上。因此,设计 XSL 文档要先从整体上考虑将整个 XSL 文档分成几个模板,然后分别设计模板的具体内容,最后将各个模板整合在一起。

4. XSL 模板定义语句

XSL 模板定义语句格式为:

```
<xsl:template match="标记名"[language="脚本语言名"]>
```

```
 ⋮
</xsl:template>
```

- match 属性用来确定哪个节点执行此模板，一般使用标记名（节点）作为属性值，其值还代表该模板名称。language 属性用来确定此模板中执行什么脚本语言，其取值与 HTML 中的 Script 标记的 Language 属性值相同，默认值是 JScript。
- XSL 模板定义语句用 match 属性指定节点，对这些特定的节点定义一种输出形式。
- XSL 文件必须有一个根模板，其 match 属性值为“/”。

5. XSL 调用模板语句

调用模板语句用来确定使用哪个模板作为输出样式。

调用模板语句格式为：

```
<xsl:apply-templates select="模板名"[order-by="标记列表"]/>
```

select 属性用来确定选择哪个模板。order-by 属性用来确定排序方式。在标记列表前添加加号（+）表示按此标记内容升序排序，减号（—）表示降序排序，标记之间以（;）分隔。

6. XSL 的循环语句

循环语句用来确定循环的节点集。

XSL 循环语句的格式为：

```
<xsl:for-each select="节点集"[order-by="标记列表"]>
    ⋮
</xsl:for-each>
```

select 属性用来确定循环体为哪个节点集。order-by 属性同上。

7. XSL 的取值语句

取值语句用来确定选择哪个节点的值。

XSL 取值语句的格式为：

```
<xsl:value-of select="节点"/>
```

select 属性用来确定选择哪个节点的值。

例 7.13 一个包含学生简历数据的 XML 文件(jl. xml)。

```
<?xml version="1.0" encoding="GB2312"?>
<?xml:stylesheet type="text/xsl" href="jl.xsl"?>
<document>
<resume>
<name>学生 A</name>
```

```
<sex>女</sex>
<birthday>1989.1.1</birthday>
<skill>唱歌</skill>
</resume>
<resume>
<name>学生 B</name>
<sex>女</sex>
<birthday>1990.12.2</birthday>
<skill>打篮球</skill>
</resume>
<resume>
<name>学生 C</name>
<sex>男</sex>
<birthday>1991.1.2</birthday>
<skill>网站开发</skill>
</resume>
</document>
```

例 7.14 显示 XML 文件中学生简历数据的 XSL 样式文件(lxr.xsl)。

```
<?xml version="1.0" encoding="GB2312"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<!--根模板-->
<xsl:template match="/">
<HTML><HEAD><TITLE>XML 技术</TITLE></HEAD>
<BODY BGCOLOR="#00CC66">
<xsl:apply-templates select="document/resume"/>
<DIV ALIGN="CENTER">
<form><input type="button" value="关闭本页" onclick="JAVASCRIPT:window.
close()"/>
</form></DIV>
</BODY></HTML>
</xsl:template>
<!--网络简历模板-->
<xsl:template match="resume">
<TABLE border="2" cellspacing="1" bordercolor="#6666FF"align="center">
<CAPTION>网络简历 [
<xsl:eval>formatIndex(childNumber(this), "I")</xsl:eval>
]</CAPTION>
<TR>
<xsl:apply-templates select="name" />
<xsl:apply-templates select="sex" />
<xsl:apply-templates select="birthday" />
</TR>
<TR><TD>技能</TD><TD COLSPAN="5">
<TABLE cellspacing="1">
```

```
<xsl:apply-templates select="skill"/>
</TABLE>
</TD>
</TR>
<TR>
<TD VALIGN="TOP">照片</TD>
<TD COLSPAN="5" ALIGN="CENTER">
<IMG SRC="girl.gif" VSPACE="2"/>
</TD>
</TR>
</TABLE>
<BR/>
</xsl:template>
<!--姓名模板-->
<xsl:template match="name"><TD>姓名</TD>
<TD><xsl:value-of/></TD>
</xsl:template>
<!--性别模板-->
<xsl:template match="sex"><TD>性别</TD>
<TD><xsl:value-of/></TD>
</xsl:template>
<!--生日模板-->
<xsl:template match="birthday"><TD>生日</TD>
<TD><xsl:value-of/></TD>
< /xsl:template>
<!--技能模板-->
<xsl:template match="skill">
<TD><xsl:value-of/></TD>
</xsl:template>
</xsl:stylesheet>
```

在浏览器中输入 http://localhost:8080/myjsp/jl.xml，可看到如图 7.8 所示页面。



图 7.8　使用 XSL 显示的简历数据

7.2.3 XSL 语言的基本语法

1. XSL 的内置函数

- value()——返回元素或属性的值。
- next()——返回集合中下一个元素。
- end()——返回集合中最后一个元素。
- index()——返回该元素在集合中的位置,返回值是一整数,如果是第一个元素返回值为 0。
- nodeName()——返回元素的名字,即标记名。
- number(参数)——将值转换为数值形式,如果不是数值则返回空。

2. XSL 的逻辑运算符

- 逻辑与——and或&&。
- 逻辑或——or或||。
- 逻辑非——not。

3. XSL 的关系运算符

- 相等——=eq。
- 相等(不区分大小写)——=ieq。
- 不等——!=ne。
- 不等(不区分大小写)——ine。
- 小于——<lt。
- 小于(不区分大小写)——ilt。
- 小于等于——<=le。
- 小于等于(不区分大小写)——ile。
- 大于——>gt。
- 大于(不区分大小写)——igt。
- 大于等于——>=ge。
- 大于等于(不区分大小写)——ige。
- 集合运算符——all,如果集合中所有项目均满足条件则返回“真”。
- 集合运算符——any,如果集合中任意项目满足条件则返回“真”。
- 集合运算符——|,返回两个集合的联合。

4. XSL 的特殊运算符

- /——选择子元素,返回左侧元素的直接子元素;如果“/”位于最左侧表示选择根结点的直接子元素。
- //——递归搜索指定的元素;如果位于最左侧表示从根结点出发递归下降搜索指

定元素。

- . ——表示当前元素。
- * ——通配符,选择任意元素,不考虑名字。
- @——取得属性值,可作为属性名的前缀。
- @* ——属性通配符,选择任意属性,不考虑名字。
- : ——分隔符,将命名空间前缀与元素或属性名分隔开来。
- !* ——在相关节点上应用指定方法。
- ()* ——分组,明确指定优先顺序。
- []——应用过滤样式。
- []* ——下标运算符,用于在集合中指示元素。

5. XML 的单选条件判断语句的应用

XML 的单选条件判断语句可以通过条件表达式的不同情况选择输出数据的样式。

例 7.15 包含数据的 XML 文档(tj.xml)。

```
<?xml version="1.0"encoding="GB2312"?>
<?xml:stylesheet type="text/xsl" href="tj.xsl"?>
<document>
<report>
<class>A组</class>
<q1>50</q1><q2>70</q2><q3>30</q3><q4>10</q4>
</report>
<report>
<class>B组</class>
<q1>20</q1><q2>30</q2><q3>40</q3><q4>50</q4>
</report>
<report>
<class>C组</class>
<q1>70</q1><q2>40</q2><q3>20</q3><q4>10</q4>
</report>
</document>
```

例 7.16 包含单选条件的 XSL 文档(tj.xsl)。

```
<?xml version="1.0" encoding="GB2312"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML><HEAD><TITLE>2008年生产统计</TITLE></HEAD>
<BODY BGCOLOR="#00CC66"><xsl:apply-templates select="document" />
<DIV ALIGN="CENTER">
<form><input type="button" value="关闭本页" onclick="JAVASCRIPT:window.
close()"/></form>
</DIV>
```

```
</BODY>
</HTML>
</xsl:template>
<xsl:template match="document">
<H3 align="center">2008 年生产产品产量统计</H3>
<TABLE border="2" cellspacing="1" bordercolor="#6666FF"align="center">
<TH>班组</TH>
<TH>一季度</TH><TH>二季度</TH><TH>三季度</TH><TH>四季度</TH>
<xsl:apply-templates select="report" />
</TABLE>
</xsl:template>
<xsl:template match="report">
<TR>
<TD><xsl:value-of select="class"/></TD>
<TD> <xsl: apply - templates select =" q1 "/> </TD > < TD > < xsl: apply - templates select =
"q2"/></TD>
<TD> <xsl: apply - templates select =" q3 "/> </TD > < TD > < xsl: apply - templates select =
"q4"/></TD>
</TR>
</xsl:template>
<xsl:template match="q1|q2|q3|q4">
<!--此处测试产量,如小于等于 20 则添加 STYLE 属性 color,其值为 red(红色)-->
<xsl:if test=".[value() $ le$ 20]">
<xsl:attribute name="style">color:red</xsl:attribute>
</xsl:if>
<xsl:value-of/>
</xsl:template>
</xsl:stylesheet>
```

在浏览器中输入 http://localhost:8080/myjsp/tj.xml,可看到如图 7.9 所示页面。



图 7.9 使用不同形式显示数据

6. XML 单选条件判断语句的结构

XML 单选条件判断语句的格式为：

```
<xsl:if expr="script-expression" language="language-name" test="pattern">
<xsl:attribute name="style">属性：属性值</xsl:attribute>
</xsl:if>
```

其中：expr 为脚本语言表达式，计算结果为“真”或“假”；如果结果为“真”，且通过 test，则按指定样式属性输出元素内容。test 为源数据，可省略该属性。

7. XML 的多选条件判断语句的应用

XML 的多选条件判断语句可以通过多个条件表达式的不同情况选择输出数据的样式。

例 7.17 包含数据的 XML 文档(dx.xml)。

```
<?xml version="1.0" encoding="GB2312"?>
<?xml:stylesheet type="text/xsl" href="dx.xsl"?>
<document>
<grade>
<name>张三</name>
<english>80</english><math>90</math><chymest>90</chymest>
</grade>
<grade>
<name>刘力</name>
<english>98</english><math>70</math><chymest>85</chymest>
</grade>
</document>
```

例 7.18 包含单选条件的 XSL 文档(dx.xsl)。

```
<?xml version="1.0" encoding="GB2312"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML><HEAD><TITLE>学生成绩</TITLE></HEAD>
<BODY BGCOLOR="#00CC66"><xsl:apply-templates select="document"/>
<DIV ALIGN="CENTER">
<form><input type="button" value="关闭本页" onclick="JAVASCRIPT:window.
close()"/></form>
</DIV></BODY>
</HTML>
</xsl:template>
<xsl:template match="document">
<TABLE border="2" cellspacing="1" bordercolor="#6666FF" align="center">
<caption>学生的成绩单</caption>
```

```
<TH>姓名</TH><TH>英语</TH><TH>数学</TH><TH>化学</TH>
<xsl:apply-templates select="grade"/>
</TABLE>
</xsl:template>
<xsl:template match="grade">
<TR>
<TD><xsl:apply- templates select="name"/></TD>
<TD><xsl:apply-templates select="english"/></TD>
<TD><xsl:apply-templates select="math"/></TD>
<TD><xsl:apply-templates select="chymest"/></TD>
</TR>
</xsl:template>
<xsl:template match="name"><xsl:value-of/></xsl:template>
<xsl:template match="english|math|chymest">
<xsl:choose>
<xsl:when test=".[value() $gt$ 85]">优秀</xsl:when>
<xsl:when test=".[value() $gt$ 70]">一般</xsl:when>
<xsl:when test=".[value() $gt$ 60]">及格</xsl:when>
<xsl:otherwise>不及格</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>
```

在浏览器中输入 http://localhost:8080/myjsp/dx.xml，可看到如图 7.10 所示页面。



图 7.10　使用不同形式显示数据

8. XML 多选条件判断语句的结构

XML 多选条件判断语句的格式为：

```
<xsl:choose>
<xsl:when expr="script - expression" language="语言名" test="pattern">数据字符
</xsl:when>
```

```
<xsl:when expr="script-expression" language="语言名" test="pattern">数据字符</xsl:when>
  ⋮
  <xsl:when expr="script-expression" language="语言名" test="pattern">数据字符</xsl:when>
  <xsl:otherwise>数据字符</xsl:otherwise>
  </xsl:choose>
```

在<xsl:choose>选择中,从第 1 个<xsl:when>开始,逐个测试 test 属性值,直到满足一个测试条件就将其中的内容输出,不再测试后面的条件;如果不满足任何一个条件,则输出<xsl:otherwise>中的内容。

在标记对<xsl:when></xsl:when>与<xsl:otherwise></xsl:otherwise>中可嵌套使用<xsl:if>或<xsl:choose>语句。

7.3 使用 JSP 获取 XML 中的数据

要处理 XML 文档,提取或修改其中的数据信息,需要先对 XML 文档进行解析,识别出 XML 文档的结构和其中的数据,解析 XML 文档需要一些程序,这些程序称为 XML 解析器,XML 文档最基本的解析器有两种,分别是文档对象模型(DOM,它是 Document Object Model 的缩写)和 XML 解析的简单 API(SAX),它们在本质上都是一种 Java 的接口。

本节主要介绍 JSP 如何使用 DOM 获取 XML 中的数据。

7.3.1 记录数据的 XML 文档

例 7.19 包含 5 条个人履历记录的 XML 文档(jspxml.xml)。

```
<?xml version="1.0" encoding="GB2312"?>
<!--个人履历表-->
<resume>
<person id="01">
    <name>张三</name>
    <birthday>03/24/1975</birthday>
    <phone>1111-1111</phone>
    <address>大连</address>
</person>
<person id="02">
    <name>李四</name>
    <birthday>9/26/1978</birthday>
    <phone>2222-2222</phone>
    <address>南京</address>
</person>
<person id="03">
```

```
    <name>王五</name>
    <birthday>11/09/1979</birthday>
    <phone>3333-3333</phone>
    <address>武汉</address>
</person>
<person id="04">
    <name>赵六</name>
    <birthday>6/04/1973</birthday>
    <phone>4444-4444</phone>
    <address>青岛</address>
</person>
<person id="05">
    <name>陈七</name>
    <birthday>12/19/1977</birthday>
    <phone>5555-5555</phone>
    <address>上海</address>
</person>
</resume>
```

7.3.2 创建包含 DOM 解析器的 JavaBean

1. 创建 JavaBean

例 7.20 包含 DOM 解析器的 Java 文档(MyDOMBean.java)。

```
package test;

import org.xml.sax.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.io.*;

public class MyDOMBean implements java.io.Serializable {
    private static String xmlStr="";
    private static final String PATH="file:///";

    public MyDOMBean() {   }

    public String getString(){ return xmlStr; }

    public static Document getDocument(String filename) throws Exception {
        xmlStr="";
        DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();
         //setIgnoringComments(true)方法指定由此代码生成的解析器将忽略注释
         dbf.setIgnoringComments(true);
```

```
        //该方法指定由此工厂创建的解析器在解析 XML 文档时,必须删除元素内容中的空格
        dbf.setIgnoringElementContentWhitespace(true);
        DocumentBuilder db=dbf.newDocumentBuilder();
        //导入 XML 文件
        Document doc=db.parse(PATH+filename);

      return doc;
  }

   public void traverseTree(Node node) throws Exception {
        if(node==null) { return;   }
        int type=node.getNodeType();                        //获得节点类型值

        switch (type) {
           //handle document nodes,当 Node.DOCUMENT_NODE=9
           case Node.DOCUMENT_NODE: {
              xmlStr+="<tr>";
              //使用 getDocumentElement()获取根元素
              traverseTree(((Document)node).getDocumentElement());
              break;
           }
           //handle element nodes,当 Node.ELEMENT_NODE=1
           case Node.ELEMENT_NODE: {
              String elementName=node.getNodeName();      //获取元素名称
              if(elementName.equals("person"))            //如果是根元素
                  {
                xmlStr+="</tr><tr>";
              }
              //获取指定子元素列表
              NodeList childNodes=node.getChildNodes();
              if(childNodes!=null) {
                 int length=childNodes.getLength();       //子节点的个数
                 for (int loopIndex=0; loopIndex<length;loopIndex++)
                 {
                    traverseTree(childNodes.item(loopIndex));
                                                          //获取子元素列表中的元素
                 }
              }
              break;
           }
           //handle text nodes,当 Node.TEXT_NODE=3
           case Node.TEXT_NODE: {
              //获取节点的值,trim()用来返回字符串并去掉前后空白
              String data=node.getNodeValue().trim();
```

```
                    if((data.indexOf(" ")<0) && (data.length()>0)) {
                      xmlStr+="<td>"+data+"</td>";
                    }
                }
            }
        }

    }
```

MyDOMBean.java 创建了一个类 MyDOMBean，它实现了 java.io.Serializable 接口可以启用其序列号化功能。

它包含 2 个具有返回值的方法，一个是 getString()方法返回字符串数据，一个是 getDocument()方法返回 Document 对象。还有一个 traverseTree(Node node)方法用来获取 XML 文档中各个节点的值。

2. 知识点归纳

1）引入需要的包

因为所创建 JavaBean 需要使用 Sun 所提供的 XML 解析器，因而需要引入 javax.xml.parsers 包，其中包含了 DOM 解析器和 SAX 解析器的程序。还要引入 org.w3c.dom 包，其中定义了 w3c 所制定的 DOM 接口。在 Java 文档的开头引入了这两个包。

```
import javax.xml.parsers.*;
import org.w3c.dom.*;
```

2）创建解析器工厂与解析器对象

使用 DocumentBuilderFacotry 类是为了创建一个解析器工厂对象，然后根据解析器工厂创建出解析器对象，使用 DocumentBuilderFactory 类的静态方法 newInstance()可以创建出一个解析器工厂对象。通过 DocumentBuilderFactory 类的静态方法 newDocumentBuilder()方法可以创建出一个具体的解析器对象。

```
DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();
DocumentBuilder db=dbf.newDocumentBuilder();
```

3）创建解析 XML 文档的 DOM 对象树

通过解析器对象的 parse(PATH＋filename)方法，可以对指定的 XML 文档进行解析，并返回一个对应 XML 文档树型结构的 Document 对象，这个 Document 对象按树模型保存了 XML 文档的数据元素。

```
Document doc=db.parse(PATH+filename);
```

4）通过 Document 对象与节点类 Node 获取 DOM 对象树中的值

通过 Node 类的 getNodeType()方法返回节点类型值，节点类型主要有 3 种：DOCUMENT_NODE 节点，其值为 9；ELEMENT_NODE 节点，其值为 1；TEXT_NODE 节点，其值为 3。

通过 Document 对象的 getDocumentElement()方法可以获取根元素;通过 NodeList 类的 getChildNodes()方法可以得到子节点列表;通过 Node 类的 getNodeName()方法可以得到节点的名称;通过 Node 类的 getNodeValue()方法可以得到节点的值。

7.3.3 创建获取 XML 值的 JSP 文档

例 7.21 输出 XML 数据的 JSP 文档(jspxml.jsp)。

```
<html>
<head>
<title>使用 DOM 解析器</title>
</head>
<%@page import="org.w3c.dom.*"%>
<%@page contentType="text/html;charset=GB2312" %>
<body bgcolor="#CFF1E1">
<center>
<h2>个人履历表(DOM 版)</h2>
<table border="1" width="80%">
<tr>
<td>姓名</td>
<td>出生年月</td>
<td>电话号码</td>
<td>居住地</td>
</tr>
<jsp:useBean id="domparser" class="test.MyDOMBean" />
<%
   Document doc=domparser.getDocument(request.getRealPath("/")+"myjsp/jspxml.xml");
   domparser.traverseTree(doc);
   out.print(domparser.getString());
%>
</body>
</html>
```

在浏览器中输入 http://localhost:8080/myjsp/jspxml.jsp,可看到如图 7.11 所示页面。

jspxml.jsp 很简单,它只是调用了 MyDOMBean 对象的三个方法,通过 getDocument(filename)方法得到一个对应 XML 树型结构的 Document 树型对象,它将 XML 数据读入树型结构并存储在内存中;通过 traverseTree(doc)方法将 XML 数据转换为字符串;通过 out.print 方法在 JSP 页面上显示用 getString()方法得到 XML 数据字符串。

当 XML 文档较大时,使用 DOM 解析器解析速度会慢,效率低。这时可以使用 SAX 解析器,它在解析 XML 文档时,从 XML 文档开始位置进行解析,同时会根据定义好的事件处理器对象判断当前所解析的成分(元素、属性或字符数据)是否需要记录与存储。

SAX 解析器在 javax.xml.parsers 包中,有兴趣的读者可以通过 SAXParserFactory



图 7.11　通过 JSP 输出 XML 文档数据

与 SAXParser 类创建一个具体的 SAX 解析器。

7.4　使用 JDOM 处理 XML 中的数据

JDOM 是一个开源项目，它基于树型结构，利用纯 Java 技术对 XML 文档实现解析、生成、序列化等操作。JDOM 利用 Java 的特性，把 ASX 和 DOM 的功能有效地结合起来。因此，只要有一个 JDOM 的 jar 包，就可以在 Java 程序中，使用 JDOM 来实现对 XML 数据的处理。

本节的内容主要介绍如何在 MyEclipse 中使用 JDOM 生成 XML 文档、修改 XML 文档中的数据。

7.4.1　在 MyEclipse 中添加 JDOM 的 jar 软件包

1. 下载 JDOM 的 jar 软件包

在 JDOM 的官方网站 http://www.jdom.org/dist/binary/下载 jdom-1.1.zip 软件包，如图 7.12 所示。

2. 确定 JDOM 的 jar 软件包的位置

下载完毕后，将其解压缩，解压缩后的“jdom.jar”文件即 JDOM 的 jar 软件包在其 build 子文件夹中，如图 7.13 所示。

3. 在 MyEclipse 中加载 JDOM 的 jar 软件包

在 MyEclipse 中加载 JDOM 的 jar 软件包的操作步骤如下所述（需要先安装好 MyEclipse 软件）。

（1）在 Windows 界面选择“开始”→“所有程序”→MyEclipse 命令，打开 MyEclipse 的工作界面，如图 7.14 所示。

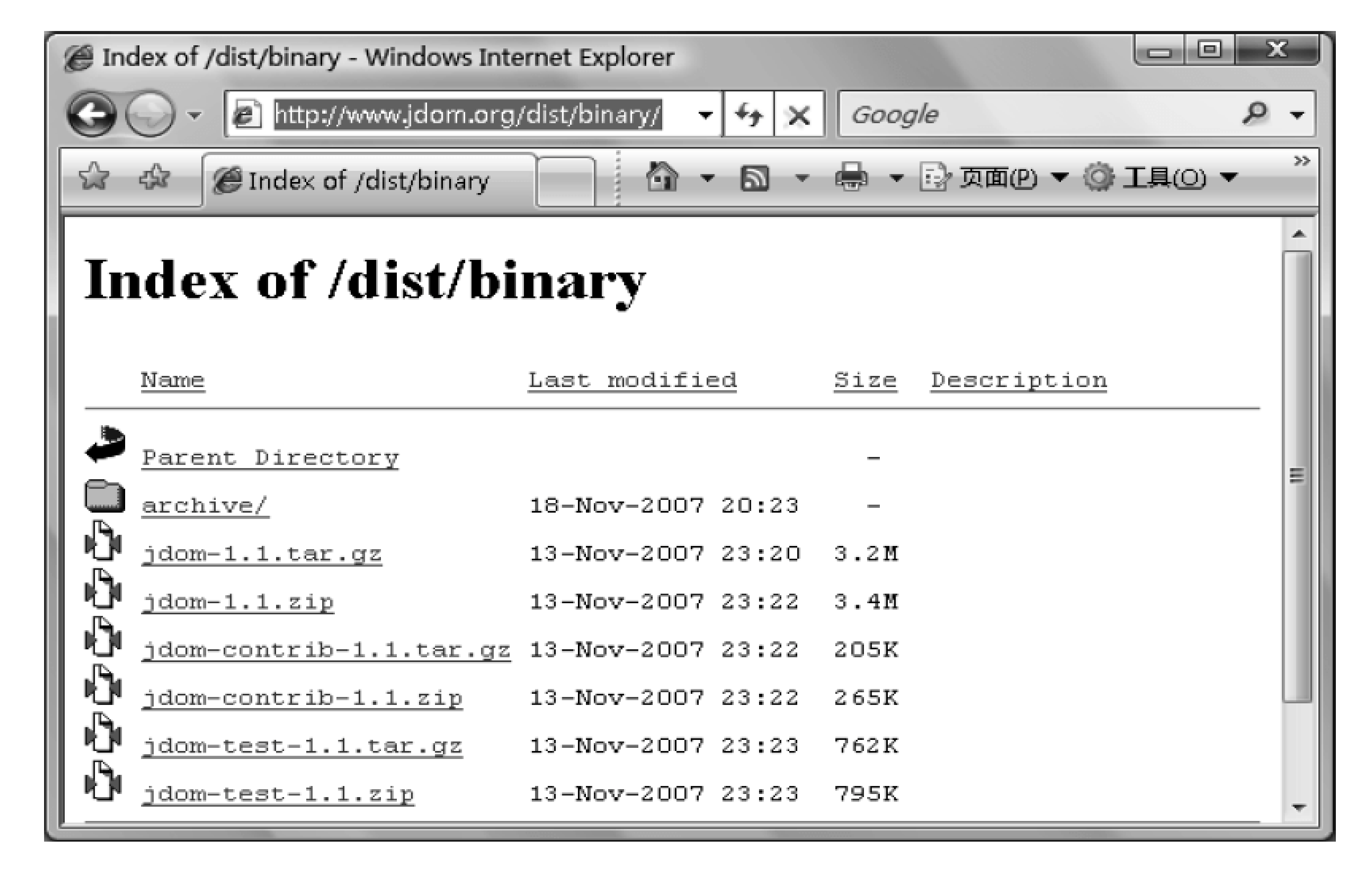


图 7.12 JDOM 的官方网站

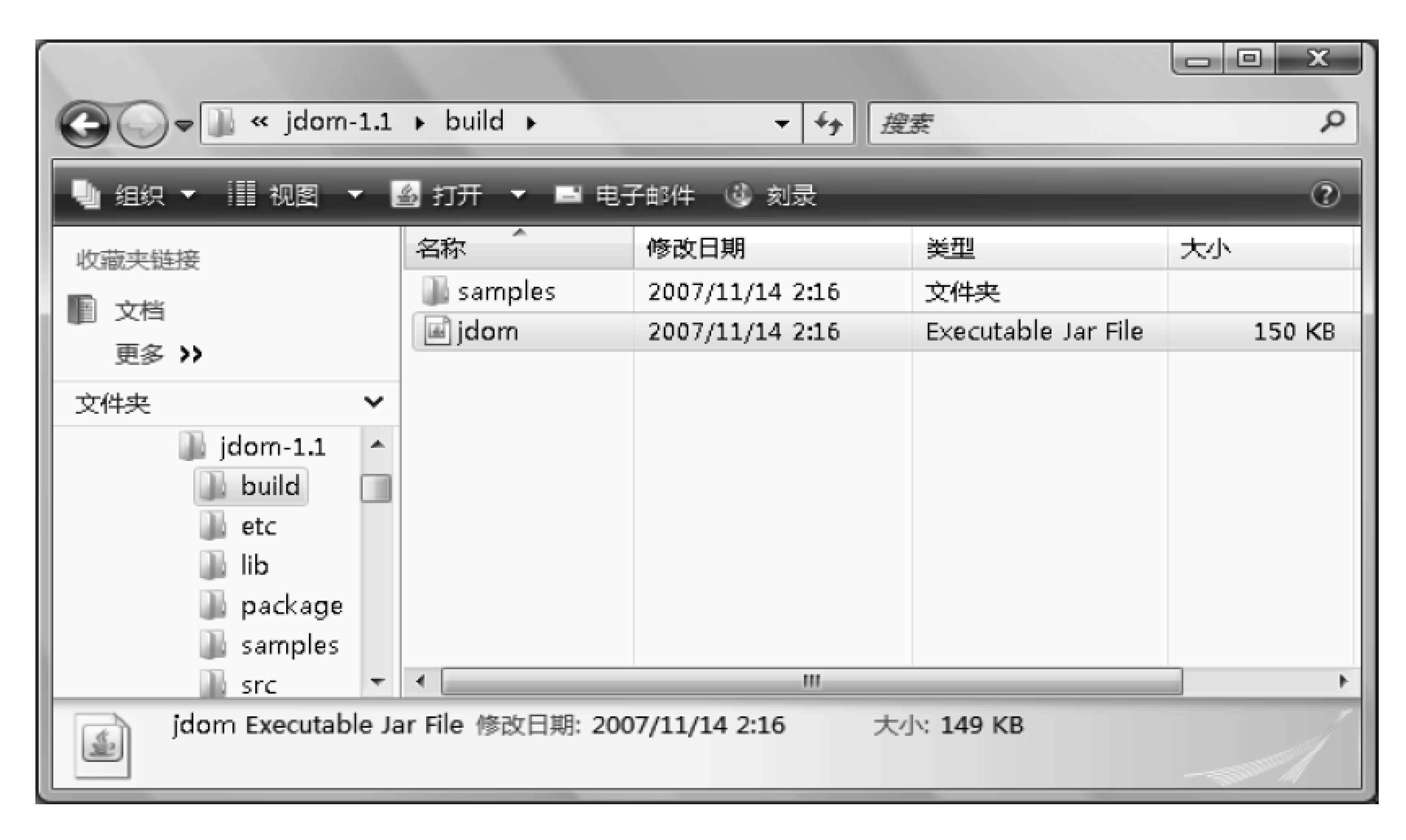


图 7.13 jdom.jar 文件目录

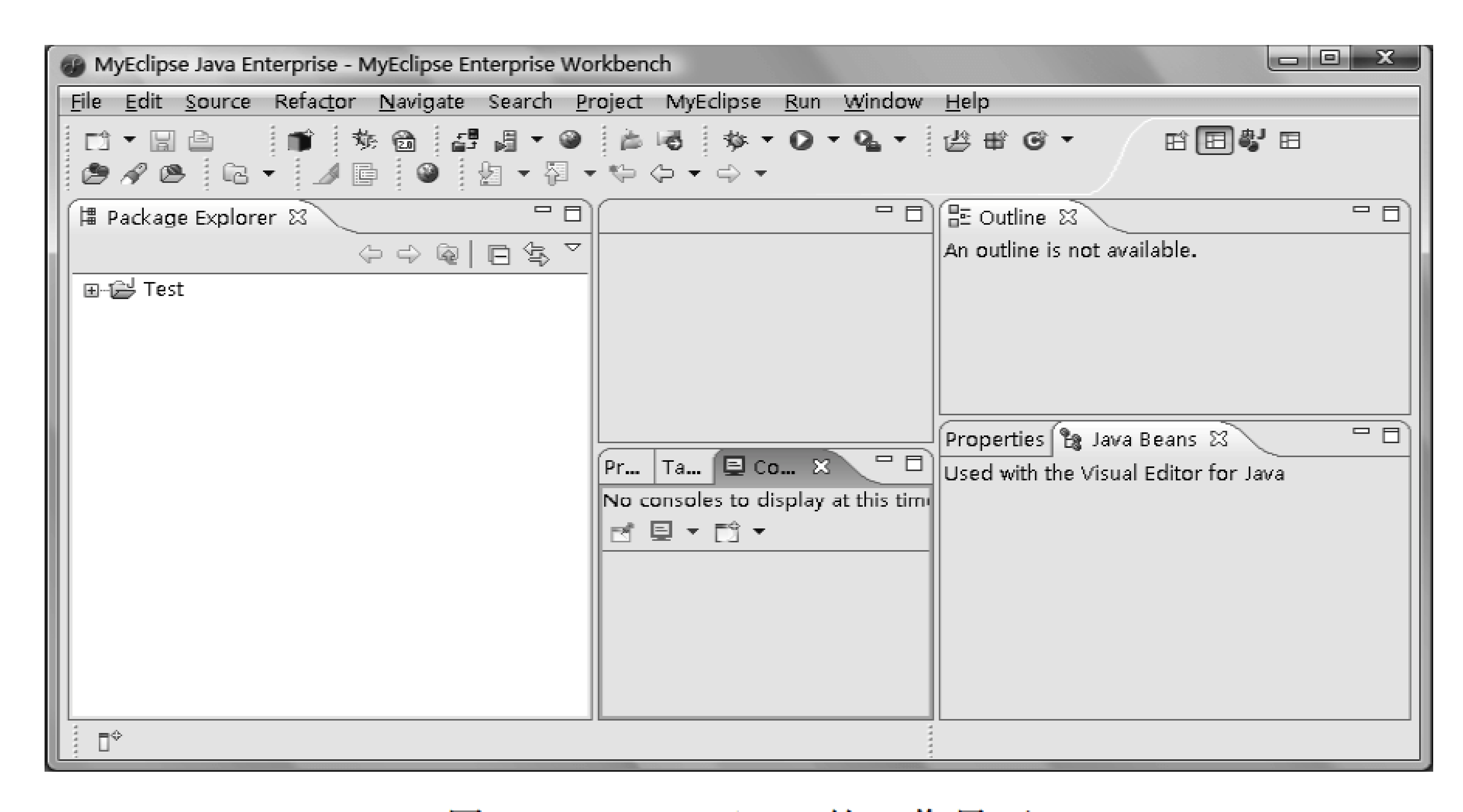


图 7.14 MyEclipse 的工作界面

(2) 在菜单栏选择 File→New→Project 命令,打开 New Project 对话框,如图 7.15 所示。从中选择 Java→Java Project,然后单击 Next 按钮。

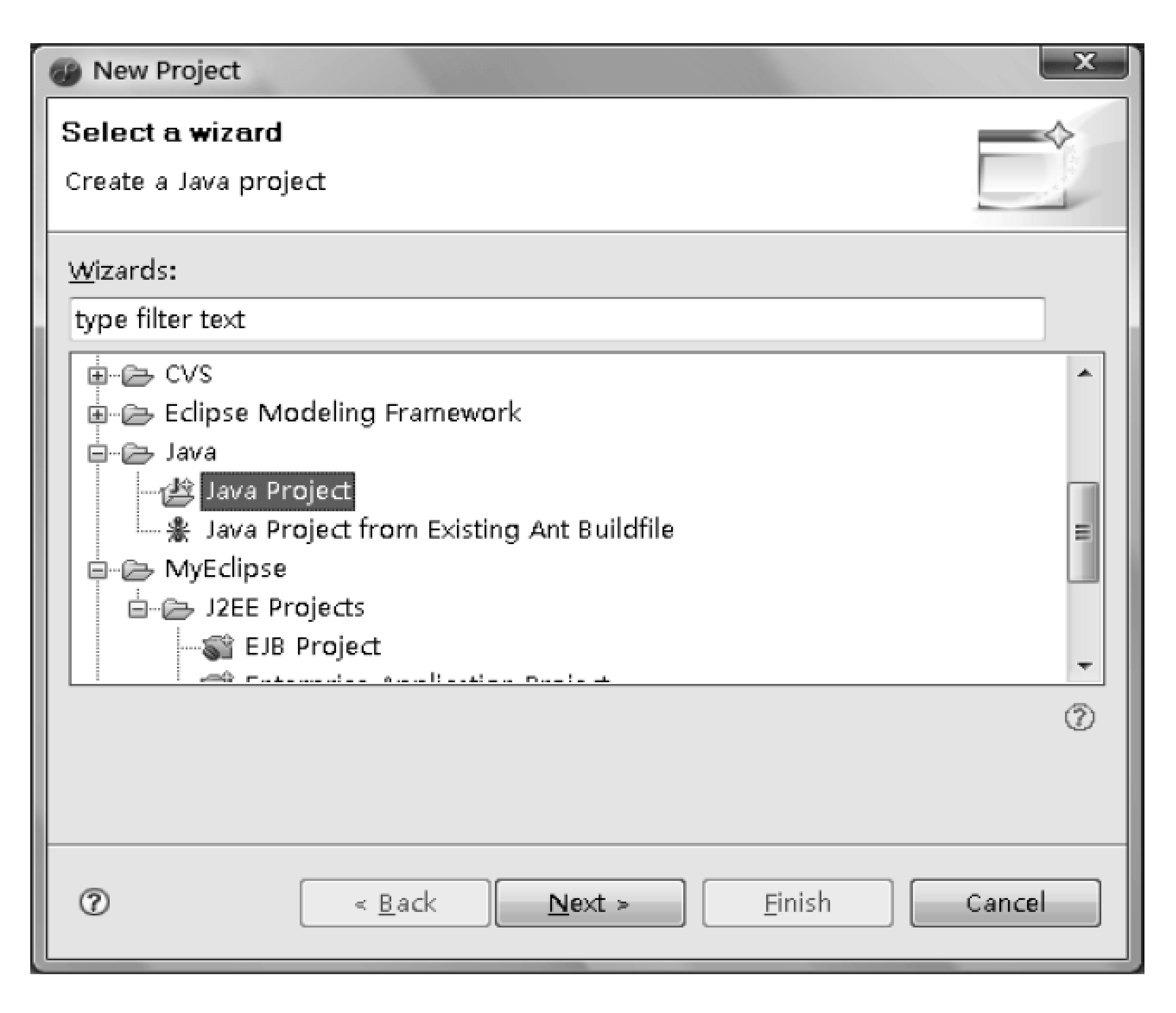


图 7.15　选择建立 Java 工程

(3) 在 New Java Project 对话框的 Project name 文本框中输入工程名称,例如 JDOMLX,如图 7.16 所示,之后单击 Finish 按钮。在工作界面可以看到新创建的工程项目,如图 7.17 所示。

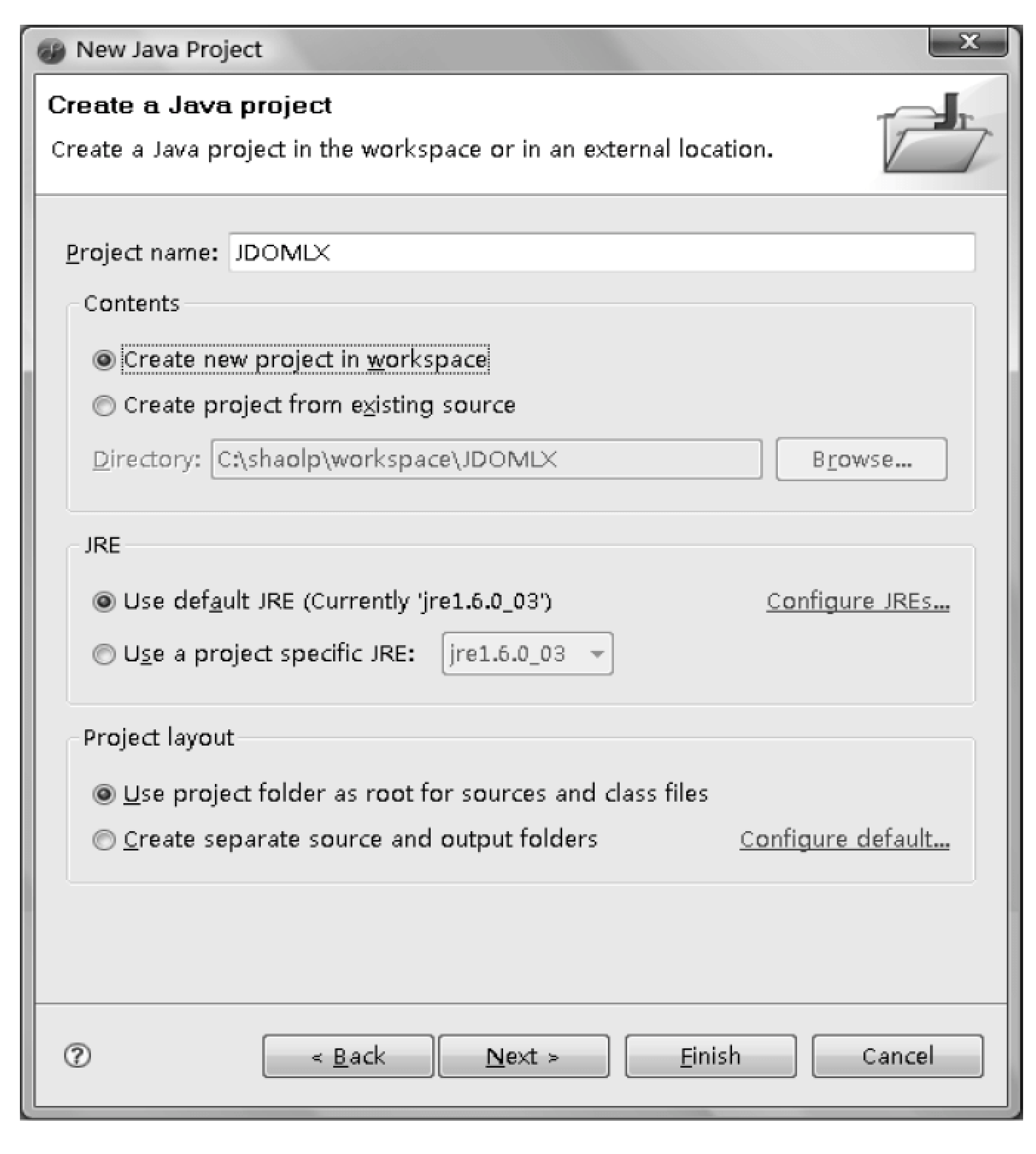


图 7.16　定义工程名称

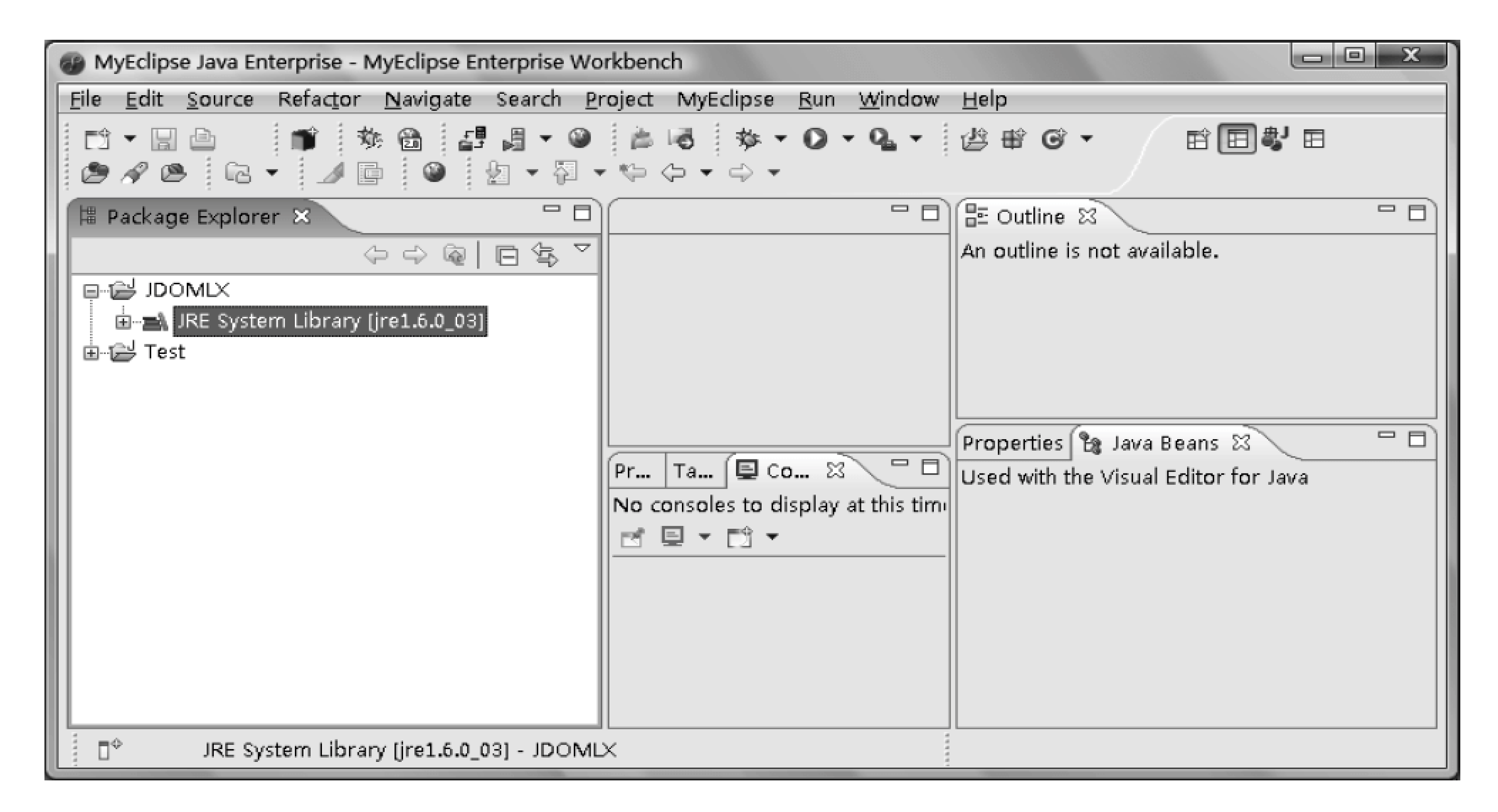


图 7.17 新创建的工程项目

(4) 右击工程名 JDOMLX,在弹出的快捷菜单中选择 Properties 菜单命令,打开 Properties for JDOMLX 对话框,如图 7.18 所示。在左边框中选择 Java Build Path 选项,在中间框中选择 Libraries 标签,在右边单击 Add External JARs 按钮,打开 JAR Selection 对话框,如图 7.19 所示,从中选择解压缩后的 jdom.jar 软件包,然后单击"打开"按钮。

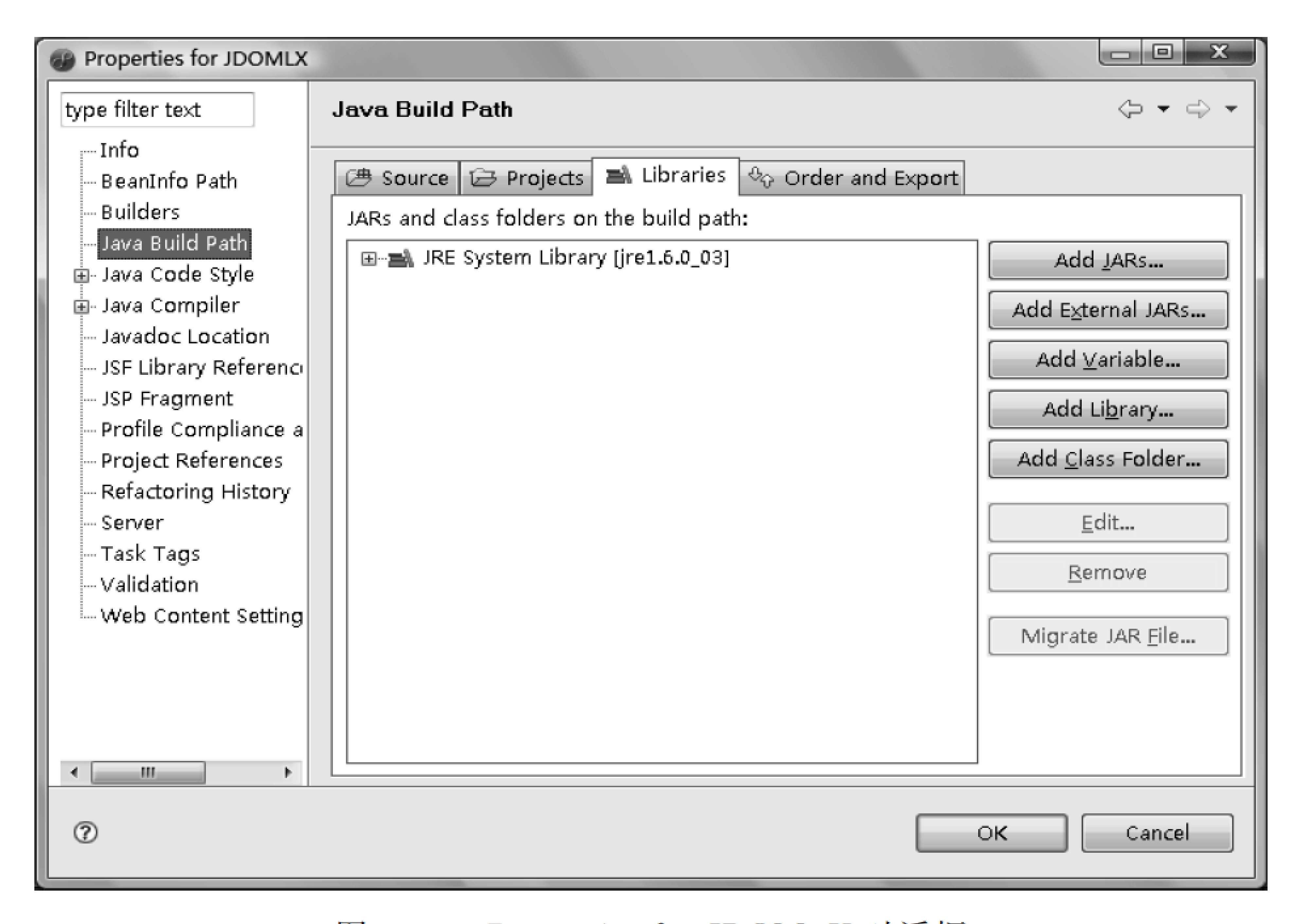


图 7.18 Properties for JDOMLX 对话框

(5) 完成以上步骤后,会在 Libraries 标签栏中看到 jdom.jar,如图 7.20 所示。说明软件包导入成功。单击 OK 按钮,结束加载 JDOM 的 jar 软件包的工作。

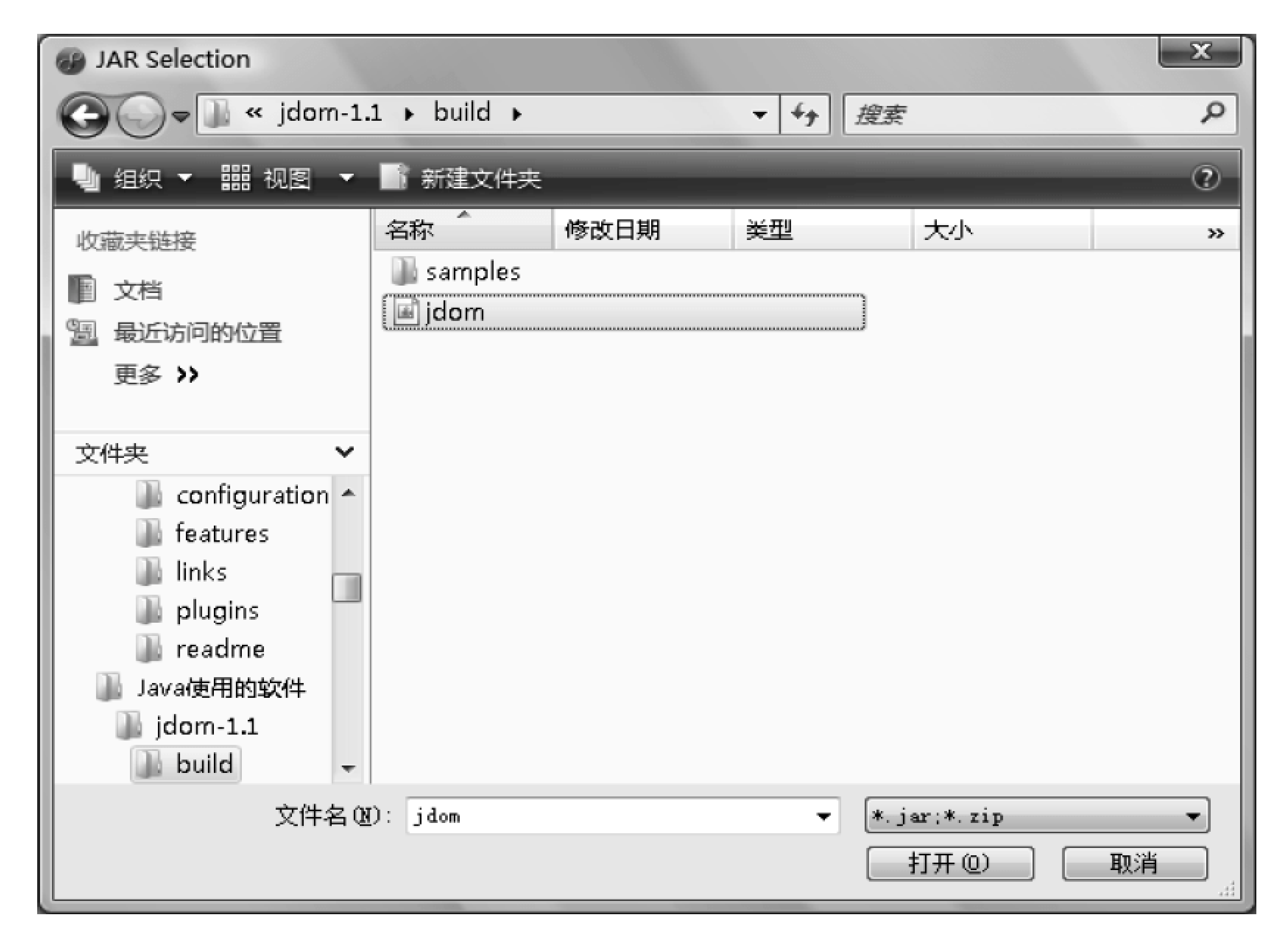


图 7.19　选择要添加的 jdom.jar 文件

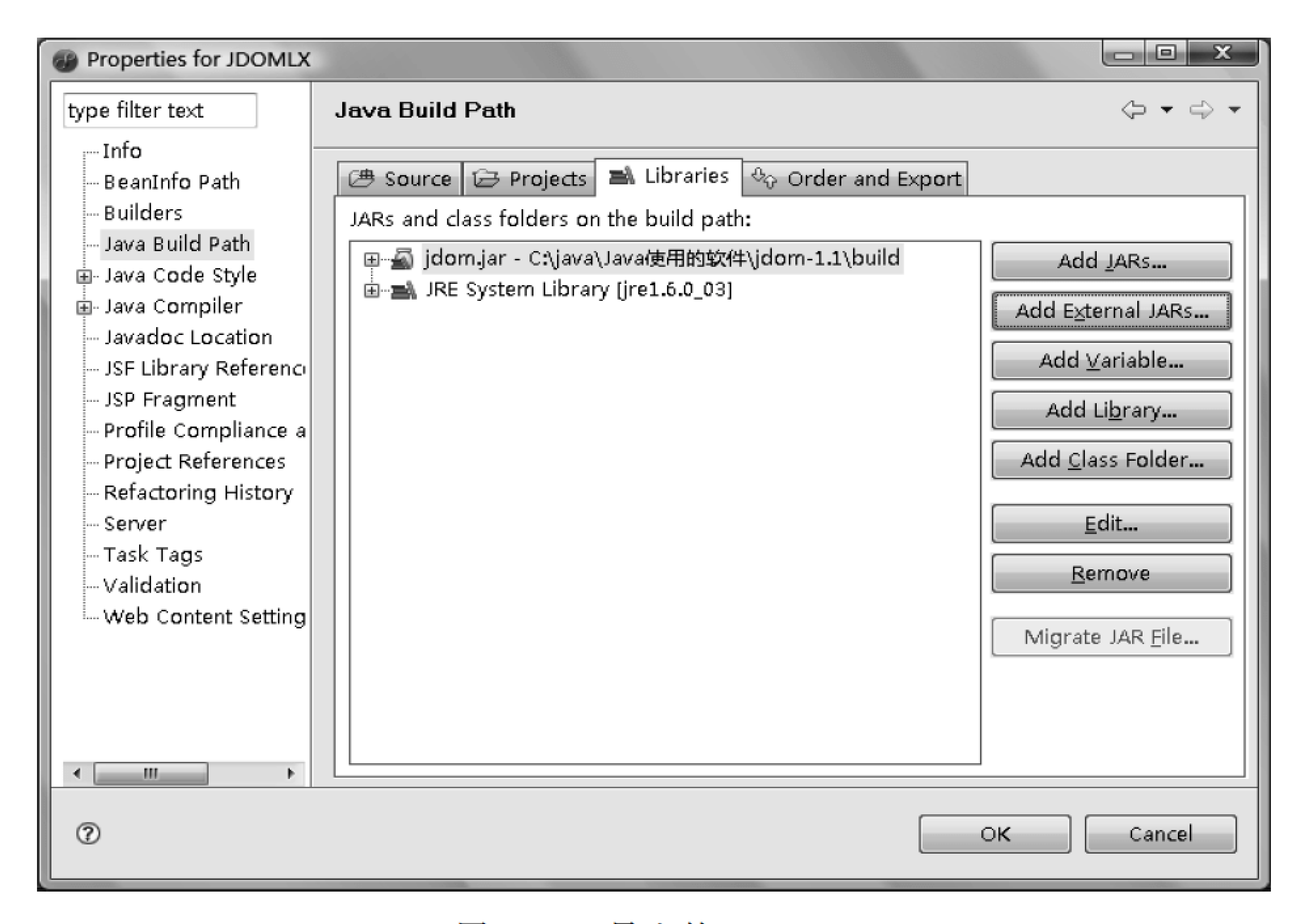


图 7.20　导入的 jdom.jar

7.4.2　使用 JDOM 生成 XML 文档

创建 XML 文档的过程如下：

(1) 在 MyEclipse 工作界面右击工程名 JDOMLX，在弹出的快捷菜单中选择 New→Class 命令，打开 New Java Class 对话框，如图 7.21 所示。在 Name 文本框输入类名

JXML，并选择 public static void main(String[] args)选项。然后单击 Finish 按钮。将打开编写 Java 程序的窗口，如图 7.22 所示。



图 7.21　New Java Class 对话框

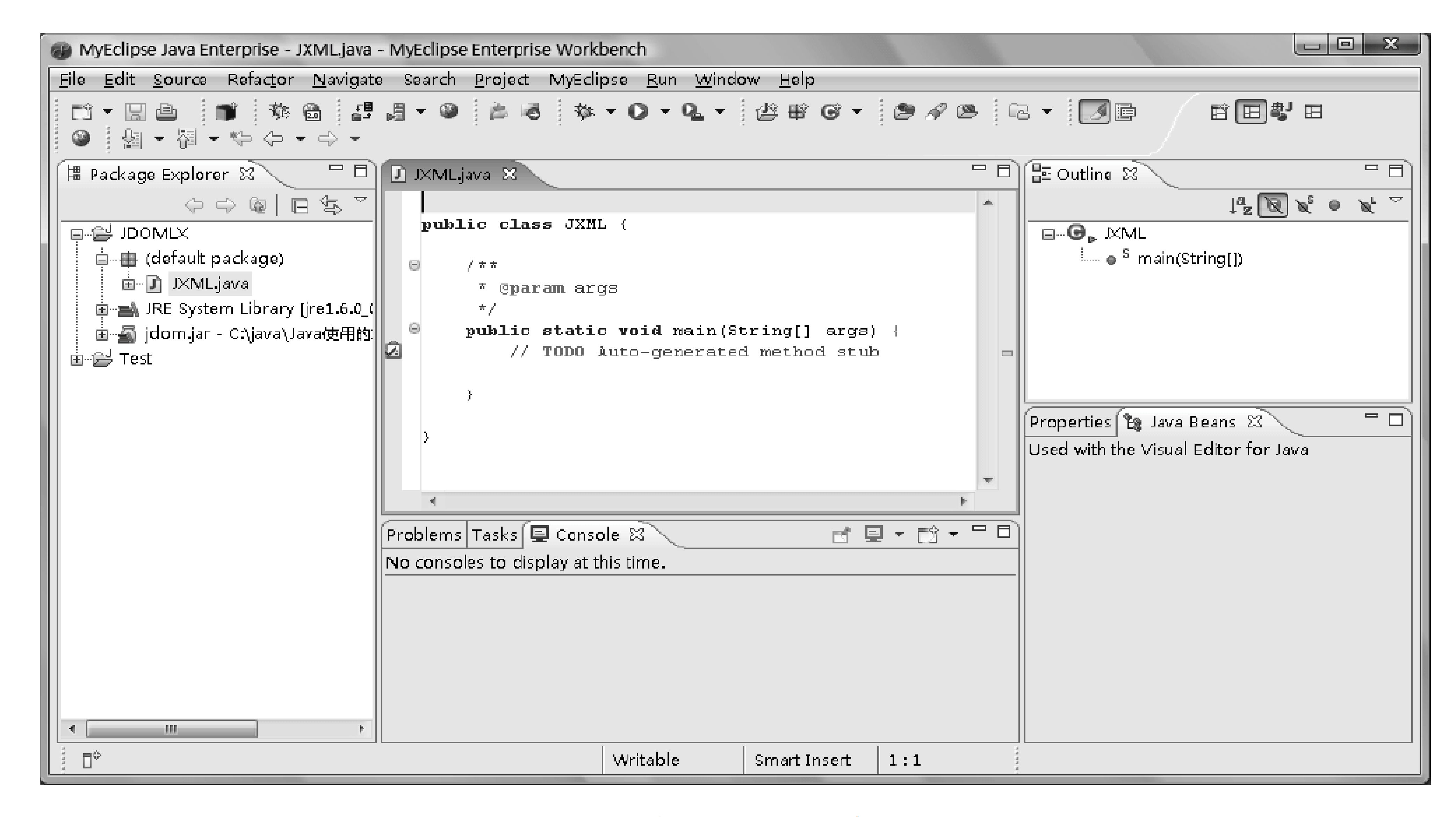


图 7.22　编写 Java 程序的窗口

(2) 在 JXML.java 视图窗口输入如下代码。

```
import java.io.* ;
import org.jdom.* ;
import org.jdom.output.* ;

public class JXML{
    //声明 XML 中所有的元素
    Element customer,name,age,sex,address,street,city,district;

    public JXML(){
        //创建 customer 元素,并设置为 XML 文档的根节点
        customer=new Element("customer");
        Document myDocument=new Document(customer);

        //创建 XML 文档中的其他元素,并初始化元素名
        name=new Element("name");
        age=new Element("age");
        sex=new Element("sex");
        address=new Element("address");
        street=new Element("street");
        city=new Element("city");
        district=new Element("district");

        //给 XML 文档中的每一个元素添加字符数据
        name.addContent("王宏");
        age.addContent("24");
        sex.addContent("男");
        street.addContent("交大东路 11 号");
        district.addContent("海淀区");
        city.addContent("北京");

        //将 street、district 和 city 元素作为子元素,添加到 address 元素中
        address.addContent(street);
        address.addContent(district);
        address.addContent(city);

        //将 address 元素作为子元素,添加到 customer 根元素中
        customer.addContent(name);
        customer.addContent(age);
        customer.addContent(sex);
        customer.addContent(address);

        try{
```

```
            //创建 XML 文件输出流
            XMLOutputter fmt=new XMLOutputter();

            //创建文件输出流
            FileWriter writer=new FileWriter("LX.xml");

            //设置所创建的 XML 文档的格式
            Format f=Format.getPrettyFormat();
            fmt.setFormat(f);

            //将生成的 XML 文档写入到"customer.xml"文件中
            fmt.output(myDocument, writer);
            writer.close();
        }
        catch(Exception e){e.printStackTrace();}
    }
    public static void main(String[]args){new JXML();}

}
```

(3) 在工具栏单击 Run 运行按钮或在工程项目中，选择 JXML.java，右击鼠标，在弹出的快捷菜单中选择 Run As → Java Application 命令，如图 7.23 所示，将在 C:\shaolp\workspace\JDOMLX 路径下看到一个生成的 LX.xml 文件，打开文件可以看到其中的内容如图 7.24 所示。

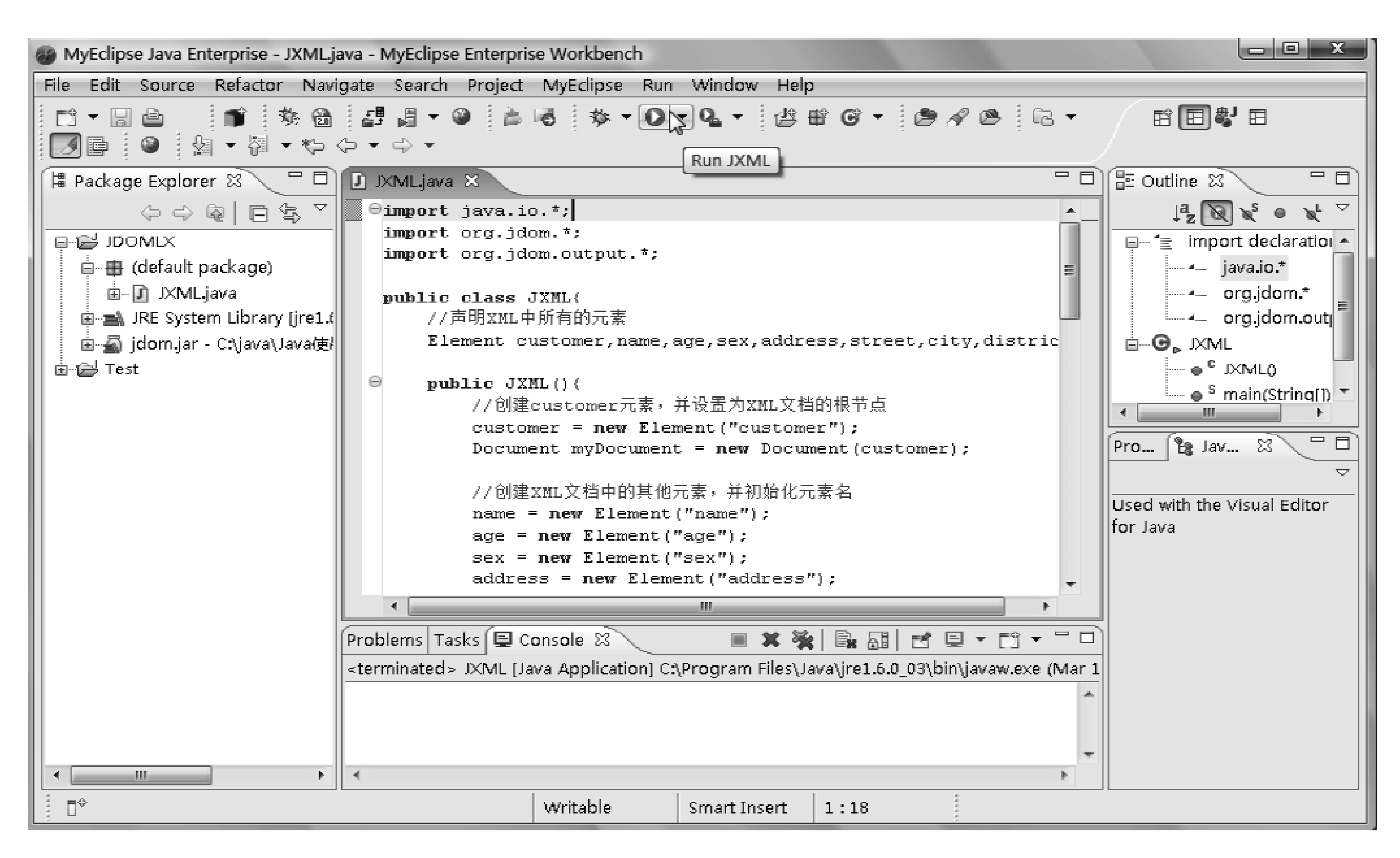


图 7.23 运行 Java 程序

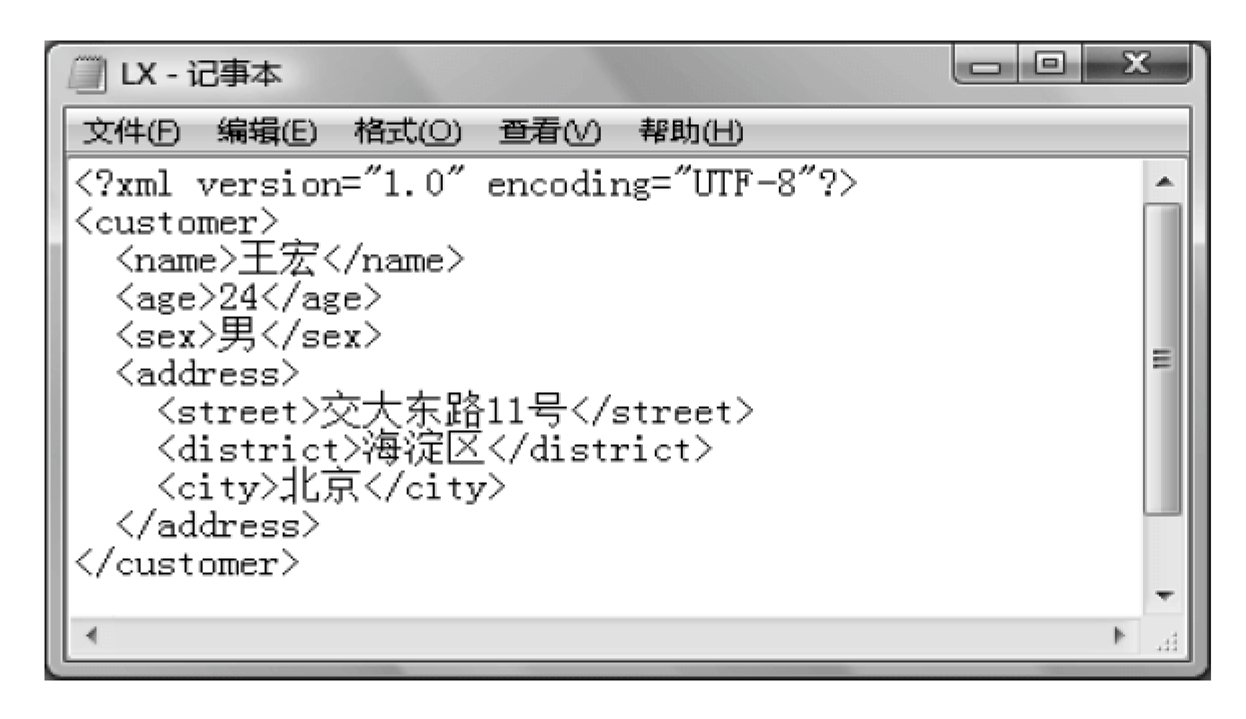


图 7.24　通过 Java 程序生成的 LX.xml 文档

7.4.3　使用 JDOM 处理 XML 中的数据

例 7.22　一个能对 XML 文档进行修改、删除、添加新元素、属性的 Java 文件（Mxml.java）。

源代码如下：

```
import org.jdom.*;
import org.jdom.output.*;
import java.io.*;
import java.util.List;
import org.jdom.input.*;
public class MXML {
    Document doc;
    List list;
    public MXML(){
        try  {
            SAXBuilder sb=new SAXBuilder();
            doc=sb.build("LX.xml");                  //通过文件构造一个 Document
            Element root=doc.getRootElement();       //得到根元素
            list=root.getChildren();        //获得 XML 文档根节点元素下的全部子节点元素

        //利用循环来处理全部子节点元素
        for(int i=0;i<list.size();i++) {
            //获得其中一个字节点元素
            Element element=(Element)list.get(i);
            String nodeName=element.getName();       //获得该子节点元素名
            //当所获得的子元素名为"address"时的处理
            if(nodeName.equals("address"))  {
                //添加子元素"country",并设置其内容为"中国"
                Element country=new Element("country");
                country.addContent("中国");
                element.addContent(country);
                //修改名为"street"的子元素内容
```

```
                Element street=element.getChild("street");
                street.setText("北京交通大学东路 11 号");
                //为 street 添加属性
                Attribute a=new Attribute("市区","true");
                street.setAttribute(a);
                element.removeChild("district");     //删除名为"district"的子元素
            }
        }
            //创建 XML 文件输出流
            XMLOutputter fmt=new XMLOutputter();
            //创建文件输出流
            FileWriter writer=new FileWriter("LX.xml");
            //设置所创建的 XML 文档的格式
            Format f=Format.getPrettyFormat();
            fmt.setFormat(f);
            fmt.output(doc, writer);                //将生成的 XML 文档写入文件中
            writer.close();
        }
        catch (Exception e){e.printStackTrace();}
    }
    public static void main(String[] args)  {new MXML();}
}
```

程序中有注释，可以了解 JDOM 的方法。

运行 Mxml.java 可以看到修改后 LX.xml 文件内容如图 7.25 所示。

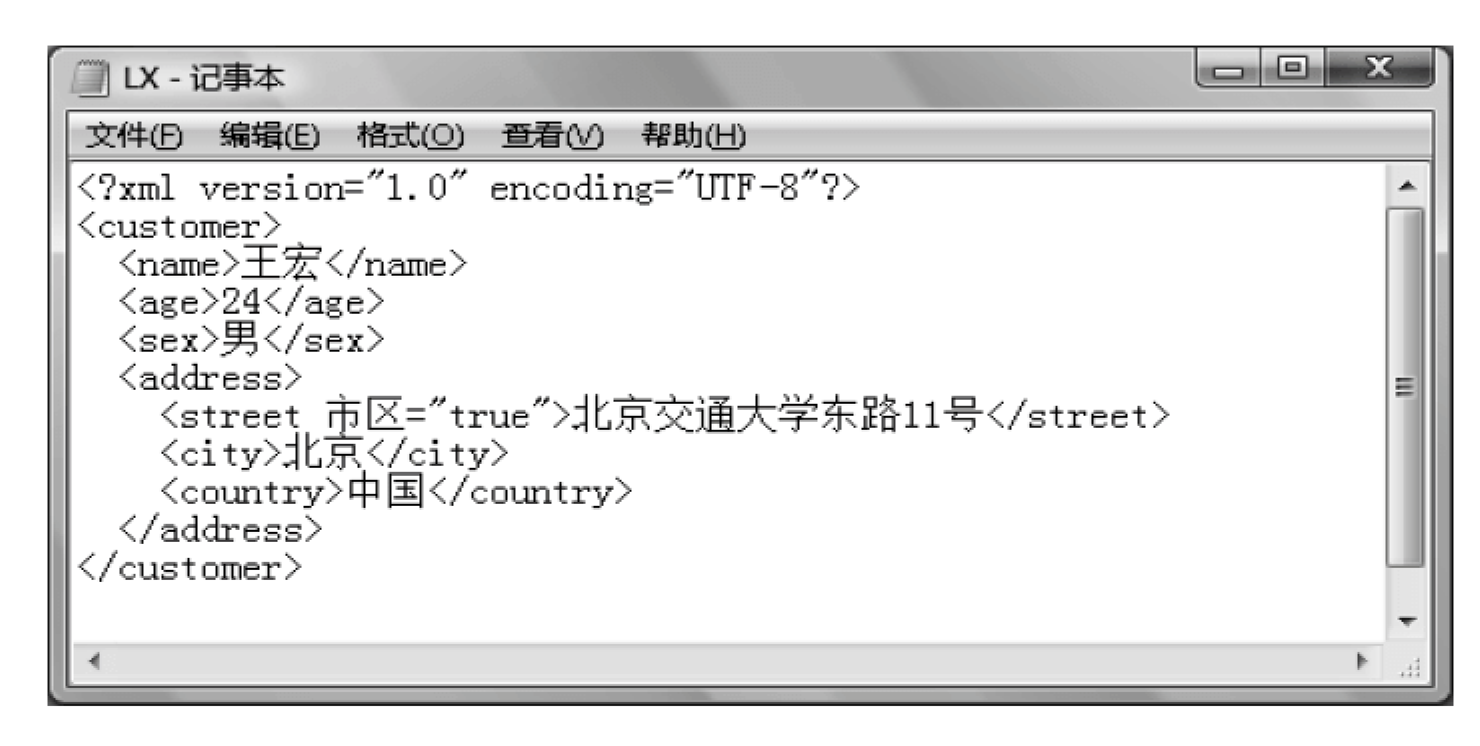


图 7.25　通过 Java 程序修改的 LX.xml 文档

XML 应用很多，例如，Tomcat 通过 conf 文件夹中的配置文件 server.xml 可以定义其服务器的各种参数，在<host></host>标记中，添加下面的语句可以增加一个虚拟目录。

```
<Context path="/lp" docBase="c:\lp" debug="0" reloadable="true"></Context>
```

在 conf 文件夹中的用户管理文件 tomcat-users.xml 中，可以看到如下内容，表示当前 tomcat 有一个用户：

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager"/>
  <role rolename="admin"/>
  <user username="lpshao" password="" roles="admin,manager"/>
</tomcat-users>
```

XML 的其他应用及用途,希望读者通过实践去发现与体会。

7.5 思考与练习

7.5.1 思考题

7-1 XML 有什么特点?

7-2 XML 有什么作用?

7-3 XSL 样式文件有什么特点?

7-4 如何运行与使用 XML 文件?

7.5.2 上机练习

7-1 使用 FrontPage 2003 编写具有良好格式的“图书基本信息”的 XML 文档,至少包括两本图书基本信息。

7-2 使用软件工具“记事本”编写“班级同学成绩”的 XML 文档,其中要包括三门课程成绩,使用 FrontPage 2003 检查文档,使其具有良好格式。

7-3 为编写好的“班级同学成绩”的 XML 文档编写 XML Schema 文档,使用 FrontPage 2003 检查 XML Schema 文档,使其具有良好格式。

7-4 编写一个 CSS 文档用于在浏览器中显示“班级同学成绩”的 XML 文档。

7-5 编写三个 XSL 文档用于在浏览器中用不同方式显示“班级同学成绩”XML 文档中的数据。

7-6 编写一个 JavaBean,创建 DOM 解析器并编写根据“班级同学成绩”的 XML 文档生成 Document 对象的方法、得到 XML 中所有数据字符串的方法、得到数据字符串的方法。

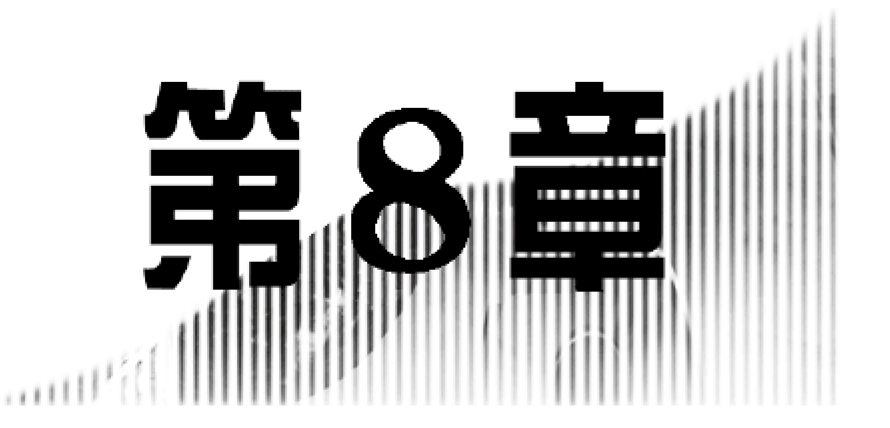
7-7 编写一个 JSP 文档,通过上题创建的 JavaBean 显示“班级同学成绩”XML 文档中的所有数据。

7-8 (选做)编写一个 JavaBean,创建 SAX 解析器并编写根据“班级同学成绩”记录大于 80 分的课程成绩,通过 JSP 文件在浏览器中显示。

7-9 下载并安装 Eclipse 与 MyEclipse,在 MyEclipse 添加 JDOM 的 jar 软件。

7-10 使用 MyEclipse,通过 JDOM 编写一个生成 XML 文档的 Java 文件。

7-11 使用 MyEclipse,通过 JDOM 编写一个读、修改、删除、添加 XML 文档中的数据的 Java 文件。



MVC 模式的 Java Web 应用程序

MVC 是一种目前广泛流行的软件设计模式，早在 20 世纪 70 年代，IBM 开始了 MVC 设计模式的研究。近来，随着 J2EE 的成熟，它正在成为在 J2EE 平台上推荐的一种设计模型，也是广大 Java 开发者非常感兴趣的设计模型。随着网络应用的快速增加，MVC 模式对于 Web 应用的开发无疑是一种非常先进的设计思想，无论选择哪种语言，无论应用多复杂，它都能在理解分析应用模型时提供最基本的分析方法，为构造的应用产品提供清晰的设计框架，为软件工程提供规范的依据。

本章主要解决以下问题：

- 什么是 MVC；
- 在 Java Web 应用中“模型”(M)模块由什么组成；
- 在 Java Web 应用中“控制器”(C)模块由什么组成；
- 在 Java Web 应用中“视图”(V)模块由什么组成；
- 如何使用 MyEclipse 编写 Java Web 应用中相关的程序文件。

8.1 什么是 MVC

MVC 是英文 Model-View-Controller 的缩写。它是解决通用的应用程序的基本方法。它不是算法而是应用程序的组织原则。

本节的内容主要介绍 MVC 的基本原则与作用。

8.1.1 MVC 的基本原则

MVC 设计模式基本原则即要求应用程序按照它们的功能进行分离，使用不同的应用程序实现数据输入、处理和输出功能。根据这个基本原则，应用程序被分成三个模块：模型(model)、视图(view)、控制器(control)。它们各自有不同的功能，由不同角色去处理各自的功能。

1. 模型、视图与控制器的功能

在 MVC 设计模式中，模型是实现控制数据访问与数据处理功能的应用程序。视

图是实现采集用户输入的数据并传递给控制器，或输出控制器中的处理数据给用户功能的应用程序。控制器是实现模型与视图之间的数据流向与数据转换功能的应用程序。

2. 模型、视图与控制器的约束条件

视图只能通过控制器从模型中取得数据；控制器要通过良好定义的、标准化的方式来与视图和模型进行通信。

按照 MVC 设计模式基本原则的要求，MVC 具备如图 8.1 所示的结构。

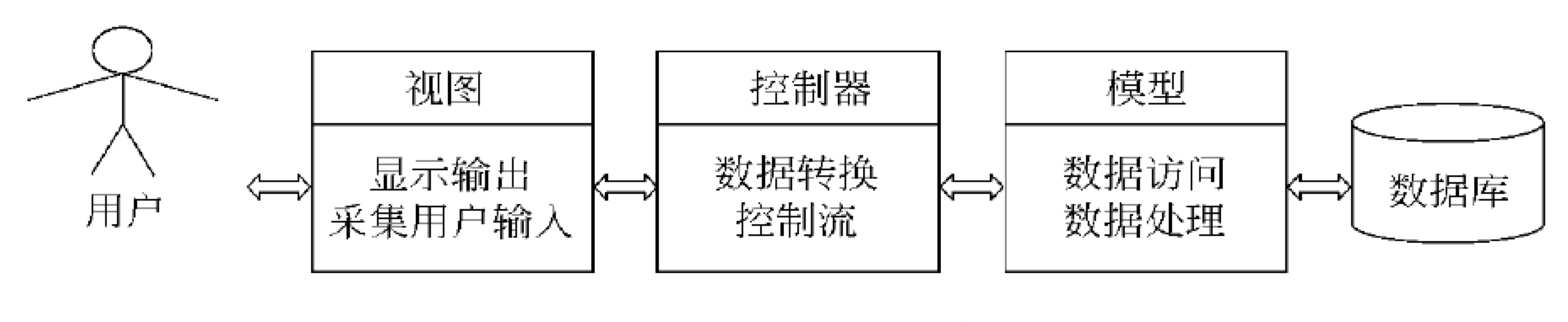


图 8.1　MVC 的结构

8.1.2　MVC 的作用

MVC 设计模式有助于将应用程序分割成若干个逻辑模块，使程序设计变得更加容易。多个视图能共享一个模型，可以使用控制器来联接不同的模型和视图去完成用户的需求，这样控制器可以为构造应用程序提供强有力的手段。给定一些可重用的模型和视图，控制器可以根据用户的需求选择模型进行处理，然后选择视图将处理结果显示给用户。使用 MVC 设计模式可以降低软件系统（模块）的耦合度，提高软件系统的可维护性与健壮性。

但使用 MVC 需要精心的计划，花费相当可观的时间去考虑如何将 MVC 运用到应用程序中，同时由于模型和视图要严格的分离，将一个应用程序分成了三个模块，所以使用 MVC 同时也意味着你将要管理比以前更多的文件，同时给调试应用程序带来了一定的困难，每个模块在使用之前都需要经过彻底的测试。因此，MVC 并不适合小型甚至中等规模的应用程序，花费大量时间将 MVC 应用到规模并不是很大的应用程序通常会得不偿失。

8.1.3　JSP/Servlet 的三层体系结构

基于 Java 技术的 Web 应用系统通常采用 Browser/Web/DataBase 的三层体系结构，如图 8.2 所示。

Web 服务器接受请求，通过应用程序服务器执行一个 Java 服务器端小程序 Servlet 并返回其输出，从而实现与客户机进行信息资源的交互。数据库服务器用来存储应用系统中所用到的各种数据，数据由数据库管理程序直接录入。应用系统的客户端只需要一个浏览器即可。用户通过浏览器来查询、增加、修改、删除数据，对数据库中的数据信息进行管理。

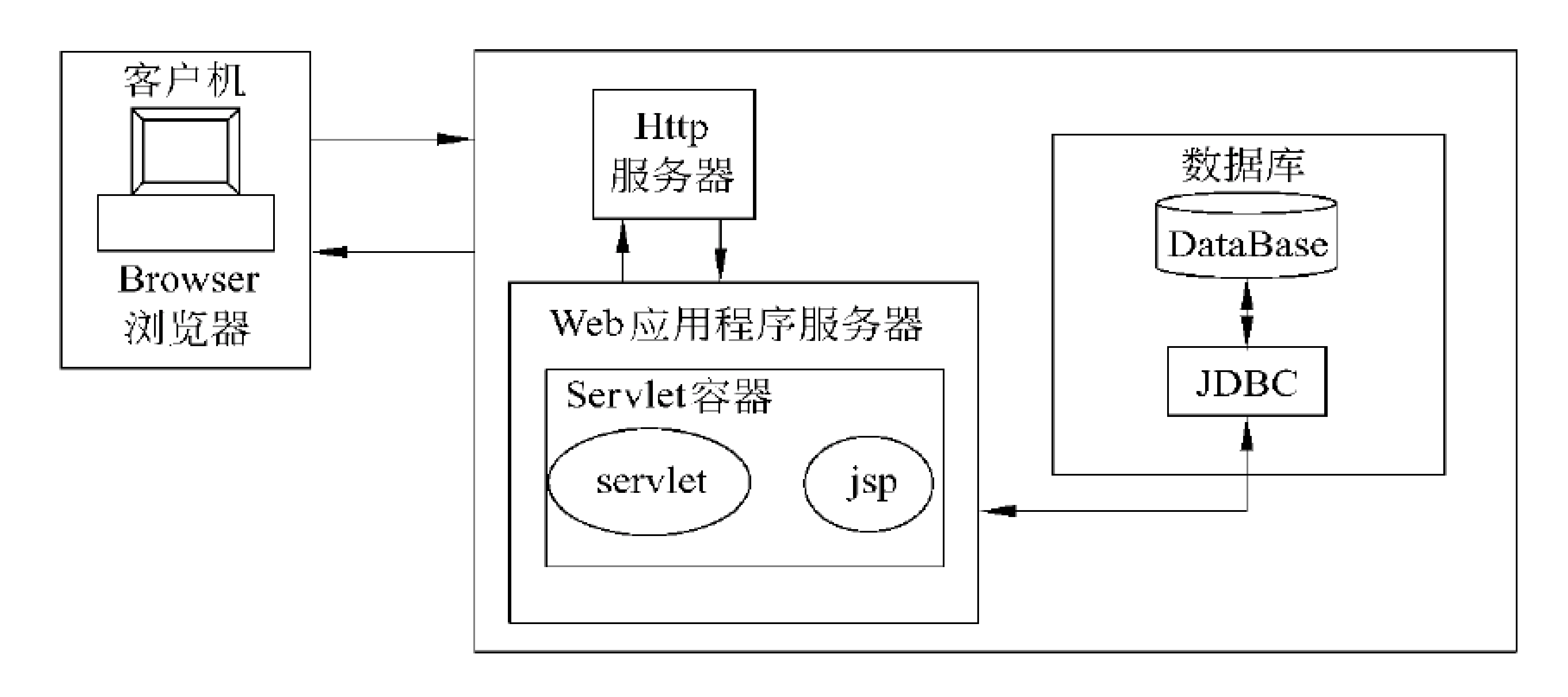


图 8.2　基于 Java 技术的 Web 应用体系结构

8.1.4　在 Browser/Web/DataBase 中间层上采用的 MVC 设计模式

在大多数 Browser/Server 结构的 Web 应用中，浏览器直接通过 HTML 或者 JSP 的形式与用户交互，响应用户的请求。虽然很直观，但是大多数应用系统操作的数据量都是惊人的，随着代码的增多会使 JSP 页面臃肿不堪，Web 服务器的负荷过重。因此，在 Browser/Web/DataBase 中间层上采用基于 MVC 的设计模式。Model 层用来实现业务逻辑，View 层用来显示用户界面，Controller 层用来负责 View 层和 Model 层之间的控制关系。具体实现时，把 Servlet 用作应用程序的控制器，把 JSP 文档作为视图，JavaBeans 被用来表示模型。所有的请求都被发送给作为控制器的 Servlet，它接受请求，并根据请求信息将它们分发给适当的 JSP 来响应。同时，Servlet 还根据 JSP 的需求生成 JavaBeans 的实例并输出给 JSP 环境。JSP 可以通过直接调用方法或使用 UseBean 的自定义标签得到 JavaBeans 中的数据。这种设计模式很好地实现了数据层与表示层的分离，使开发工作更加容易和迅速。在这种设计模式下，各层次之间的数据传递如图 8.3。

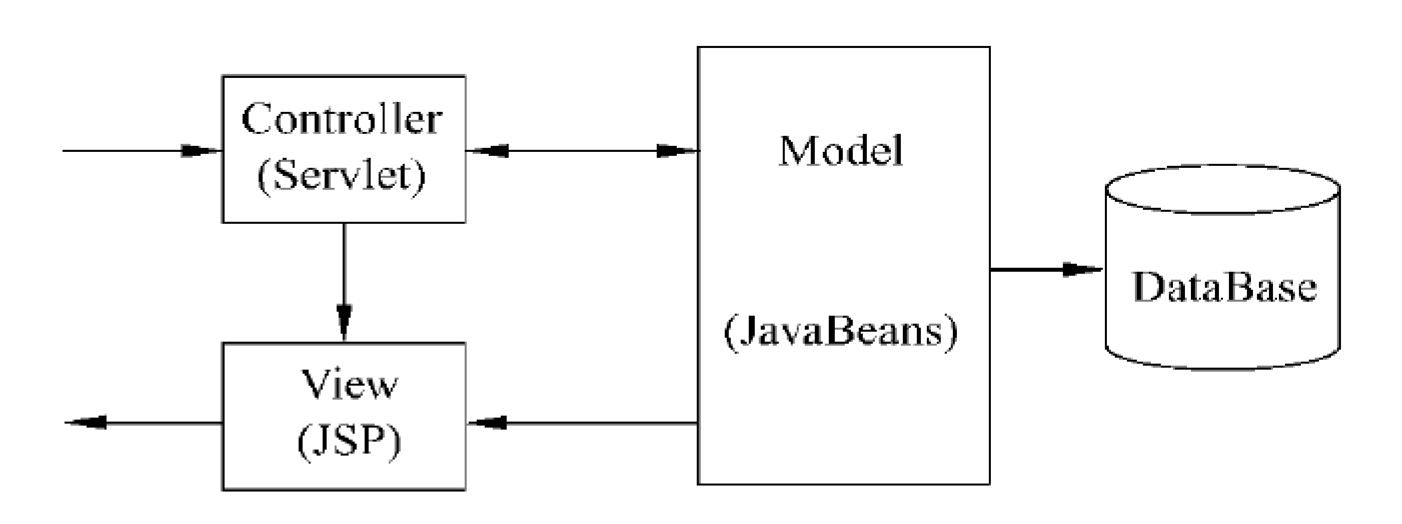


图 8.3　MVC 设计模式的数据传递图

在数据的存取方面，采用 JSP/Servlet+JDBC 的技术，即客户端不产生数据库查询命令，客户端上的浏览器通过 URL 和中间层的 Web 服务器建立连接。Web 服务器主要负责接收本地或远程浏览器的 HTTP 数据请求，然后中间层的 Servlet 收到请求后，通过执行程序中的 SQL 语句，利用 JDBC 提供的标准 API 对数据库进行访问和相应的操作处理。Servlet 再将查询的数据传给 JSP，最后生成标准的 JSP 页面将结果返回给提出请求的浏览器。这样，不仅将客户端与数据库服务器端分开，同时提高了数据库的访问效率。

8.2 采用 MVC 结构的用户登录系统

本节的内容主要介绍使用 JSP、Servlet 与 JavaBean 编写具有 MVC 设计模式的用户登录系统。

8.2.1 用户登录系统结构

1. 软件结构

用户登录系统是一个 Java Web 应用系统，采用典型 B/S 模式的“客户层”、“应用层”与“数据层”3 层软件结构。

客户层——提供基于客户端浏览器的 JSP 用户页面，即视图。

应用层——提供运行在服务器端的控制器与模型。

数据层——提供存放在服务器端的存放和维护 Web 应用中所用到的数据信息。

实现 3 层软件结构的符合 MVC 原则的用户登录系统的程序框架如图 8.4 所示。

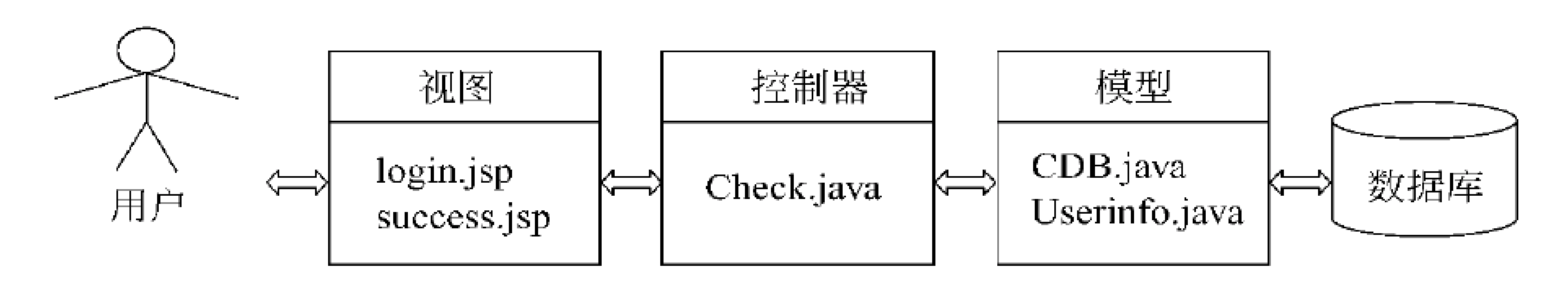


图 8.4　用户登录系统程序框架

2. 用户登录系统中主要程序文件说明

(1) 构成视图的应用程序

login. jsp——用户登录页面。

success. jsp——成功登录页面。

(2) 构成控制器的应用程序

Check. . java——用于转换用户输入数据、定义页面转向的 Servlet。

web. xml——用于定义 Servlet 位置的服务器配置文件。

(3) 构成模型的应用程序

CDB. java——用于连接数据库的 JavaBean。

CD. java——用于验证用户信息的 JavaBean。

Userinfo. java——用于临时存放从数据库提取的用户信息的 JavaBean。

(4) 存放数据的数据库文件

这里使用第 6 章中已经创建的数据库文件 questionnaire 及表对象 admins。

8.2.2 在 MyEclipse 中配置 Tomcat 服务器

在 MyEclipse 中编写 Web 应用系统的程序文件会比较方便，它可以编译 Java 文件，自动设置 JavaBean、Servlet 程序文件的存放位置，使用 Tomcat 运行 JSP 文件。

下面介绍在 MyEclipse 中配置 Tomcat 服务器的操作步骤。

(1) 启动 MyEclipse,打开 MyEclipse 工作界面,单击菜单栏 Window(窗口)→Preference…(首选项)菜单,打开 Preference (首选项)对话框,如图 8.5 所示。

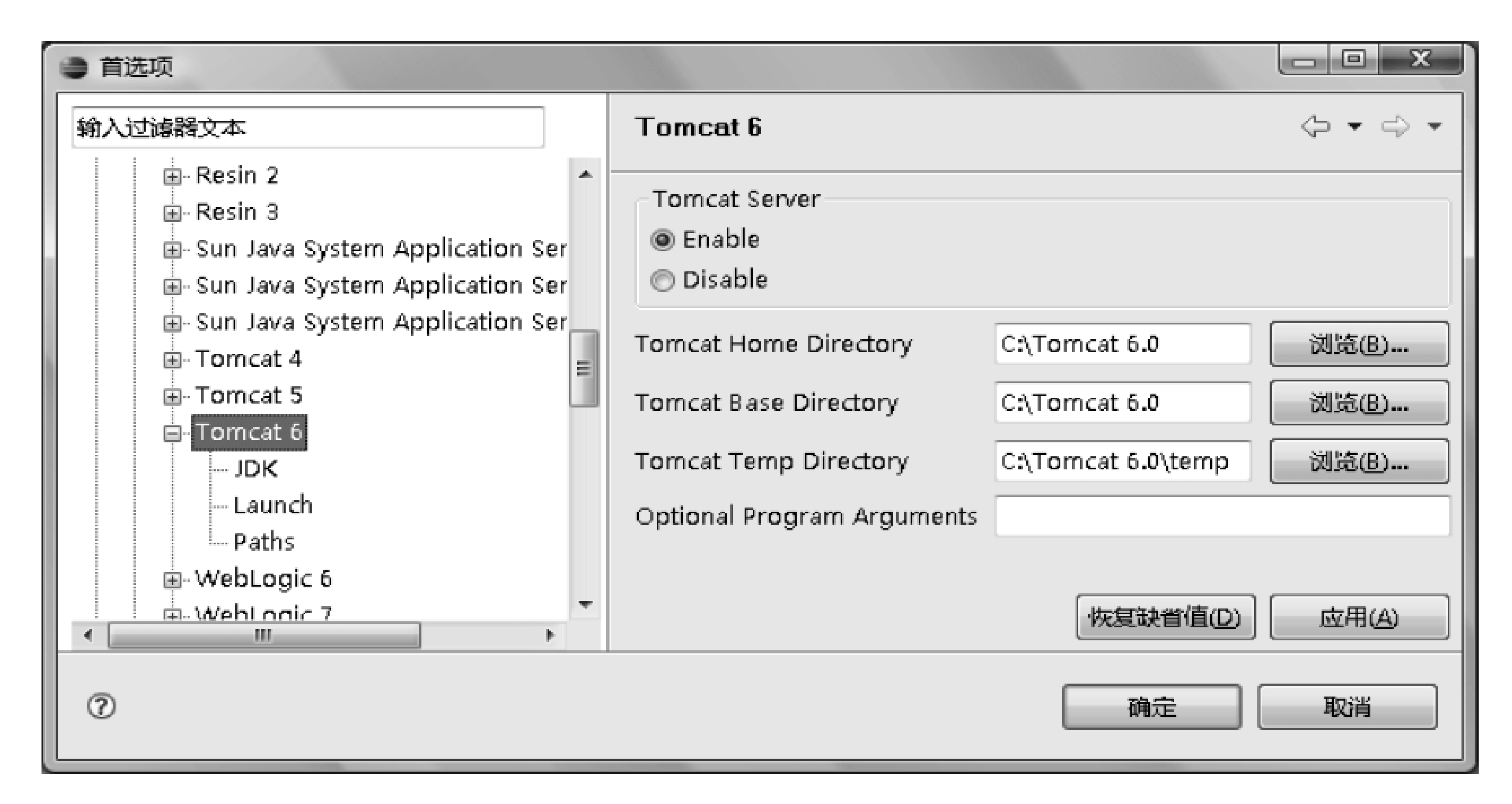


图 8.5 Preference (首选项)对话框

(2) 在左边菜单树中选择 MyEclipse→Application Servers→Tomcat 6 选项,在右边 Tomcat Server 框中选中 Enable 单选项,然后单击 Tomcat Home Directory 文本框后的 Browse(浏览)按钮,选择 Tomcat 的安装根目录,这里是 C:\Tomcat 6.0,其他文本框会自动填充。然后单击右下角的 Apply(应用)按钮,如图 8.5 所示。

(3) 在左边菜单树中选择 MyEclipse→Application Servers→Tomcat 6→JDK 选项,在 Tomcat JDK Name 框下选择 JDK 的安装路径,如图 8.6 所示。然后单击右下角的 Apply(应用)按钮,如图 8.6 所示。这里要特别注意,默认的是 JRE 的运行环境,这里要设定成 JDK 的安装路径,否则,MyEclipse 无法正常部属 Web 应用,也无法正常运行 Tomcat 服务器。

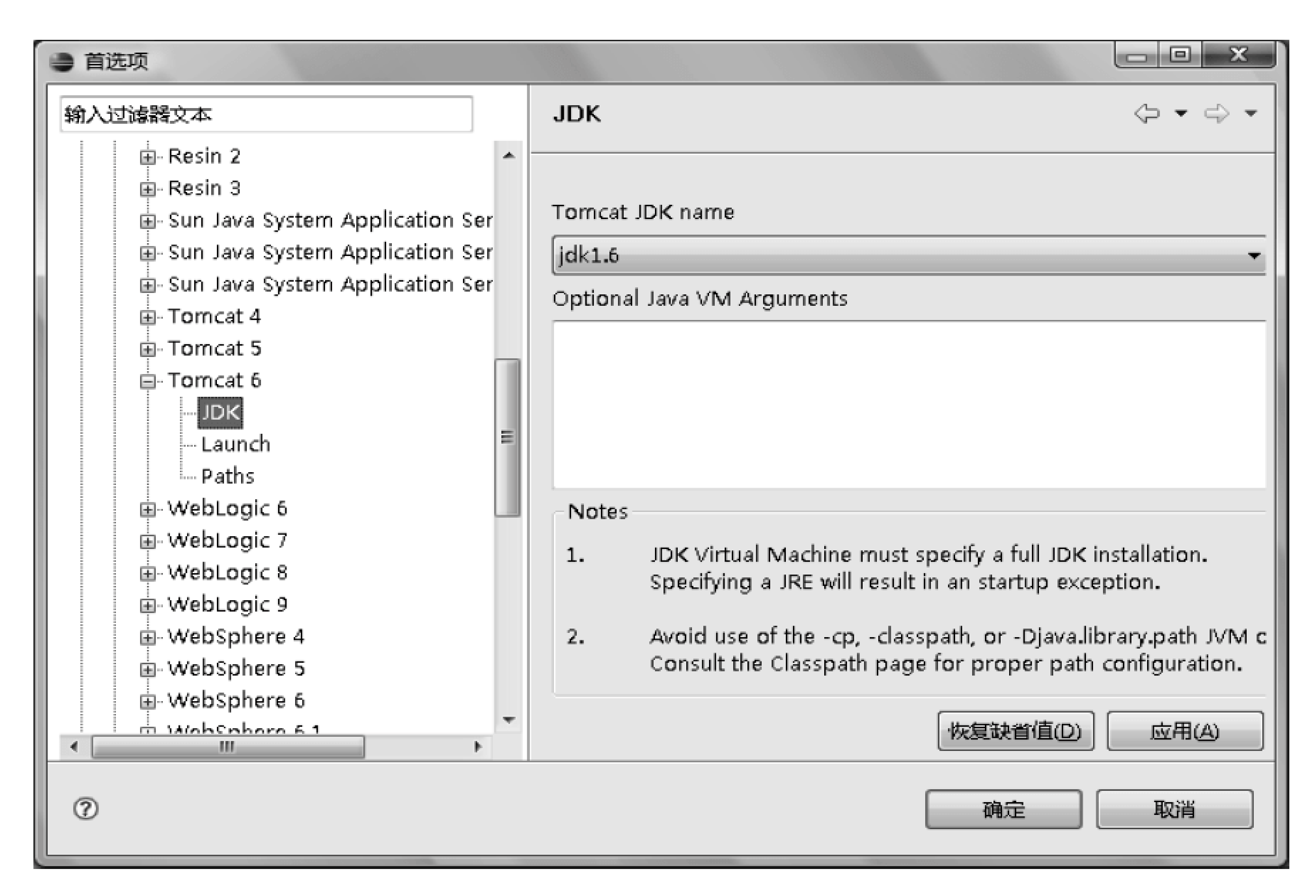


图 8.6 选择 JDK

（4）在 Preference（首选项）对话框单击 OK（确定）按钮，即可完成 Tomcat 的配置。

在 MyEclipse 工作界面单击工具栏上的“服务器”按钮，从下拉选项中可以看到添加进来的 Tomcat 6 服务器按钮，如图 8.7 所示。单击 Start 开始按钮，将启动 Tomcat 6 服务器开始工作。

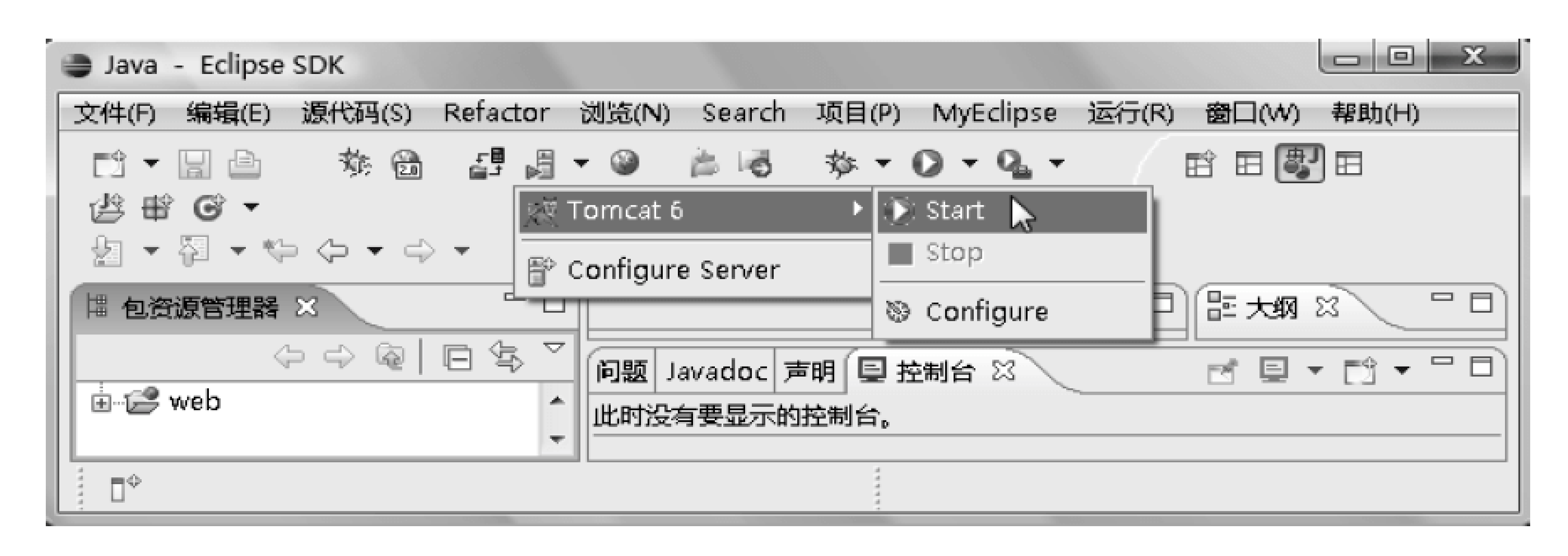


图 8.7　Tomcat 6 服务器按钮

8.2.3　设置 WZKF 工程项目

1. 创建 WZKF 工程项目

（1）在 MyEclipse 工作界面，在菜单栏单击 File（文件）→New（新建）→Project（项目）菜单命令，打开“新建项目（New Project）”对话框，从中选择 MyEclipse→J2EE Projects→Web Project，如图 8.8 所示，然后单击“下一步”按钮。

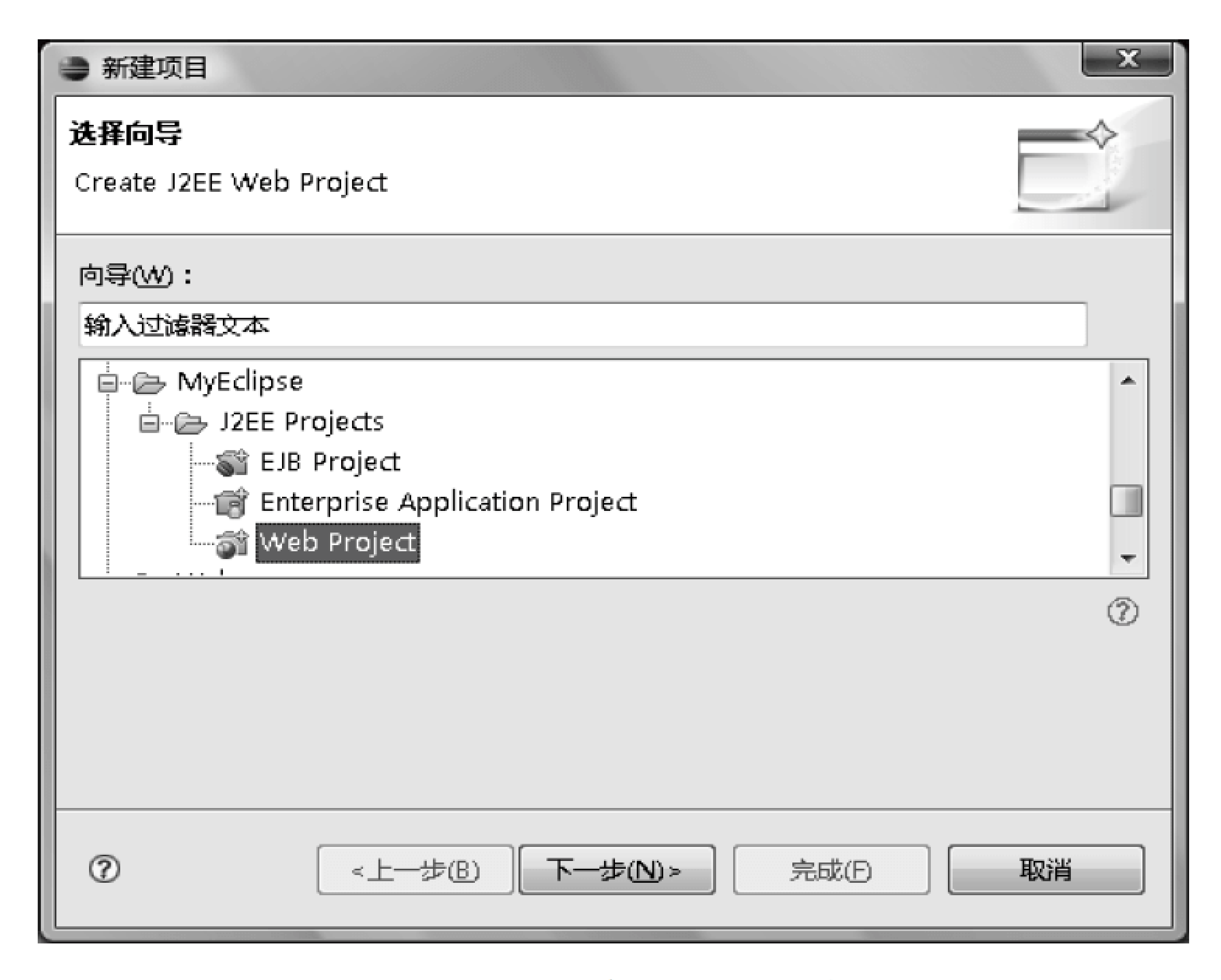


图 8.8　“新建项目”对话框

（2）在 Web Project Details 框的 Projects Name 文本框中输入 WZKF，如图 8.9 所示，其他采用默认值，注意这里 Context root URL 自动设置为/WZKF。然后单击“完成”按钮即可完成创建项目的工作。

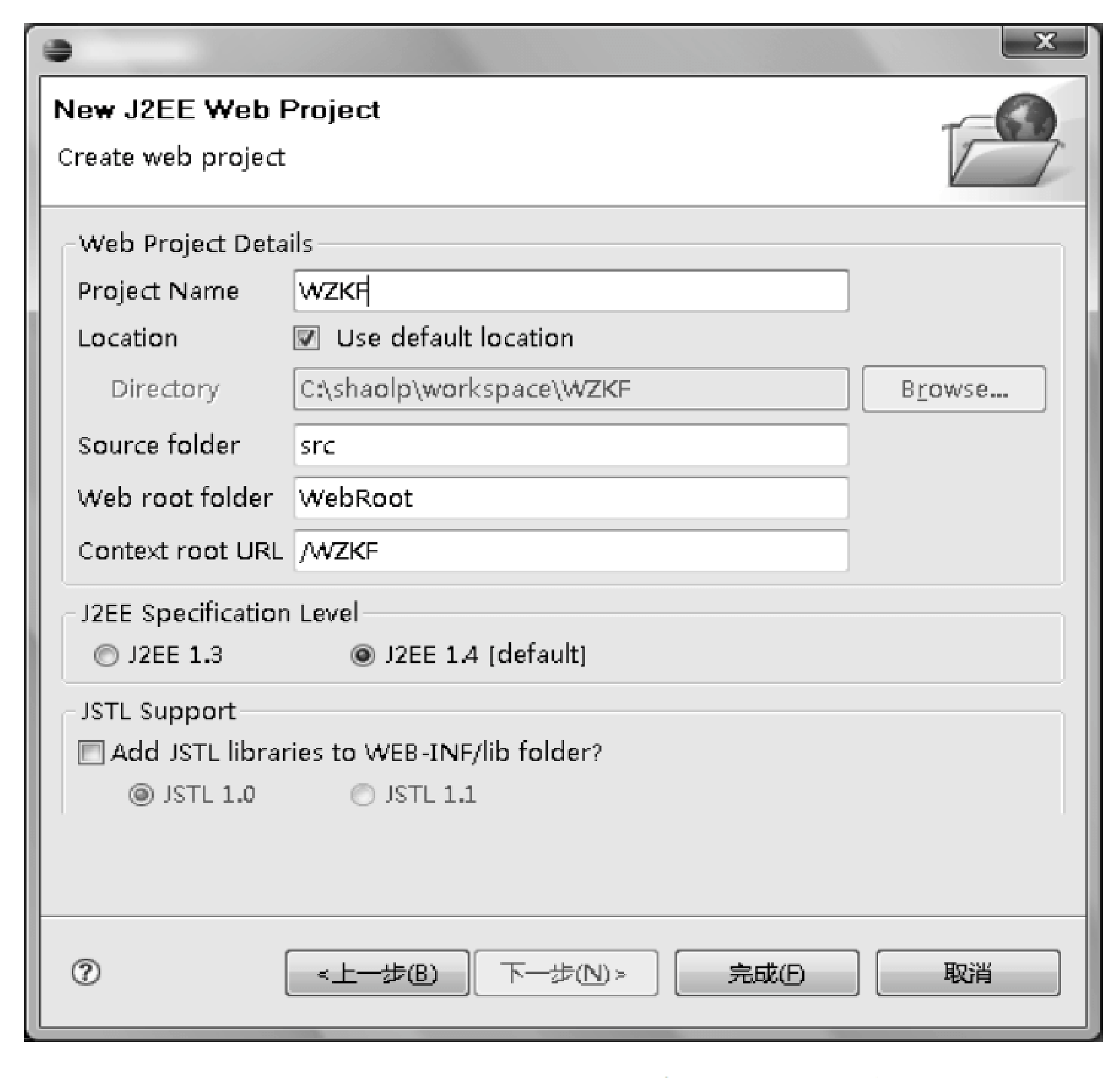


图 8.9　New Project(新建项目)对话框

2. 设置虚拟目录

(1) 在 MyEclipse 工作界面“包资源管理器”中选中 WZKF 工程项目的根目录，单击工具栏上的“部署应用程序”按钮，打开 Project Deployments 对话框，确保 Projects 框中为 WZKF，然后单击 Add 按钮，如图 8.10 所示。

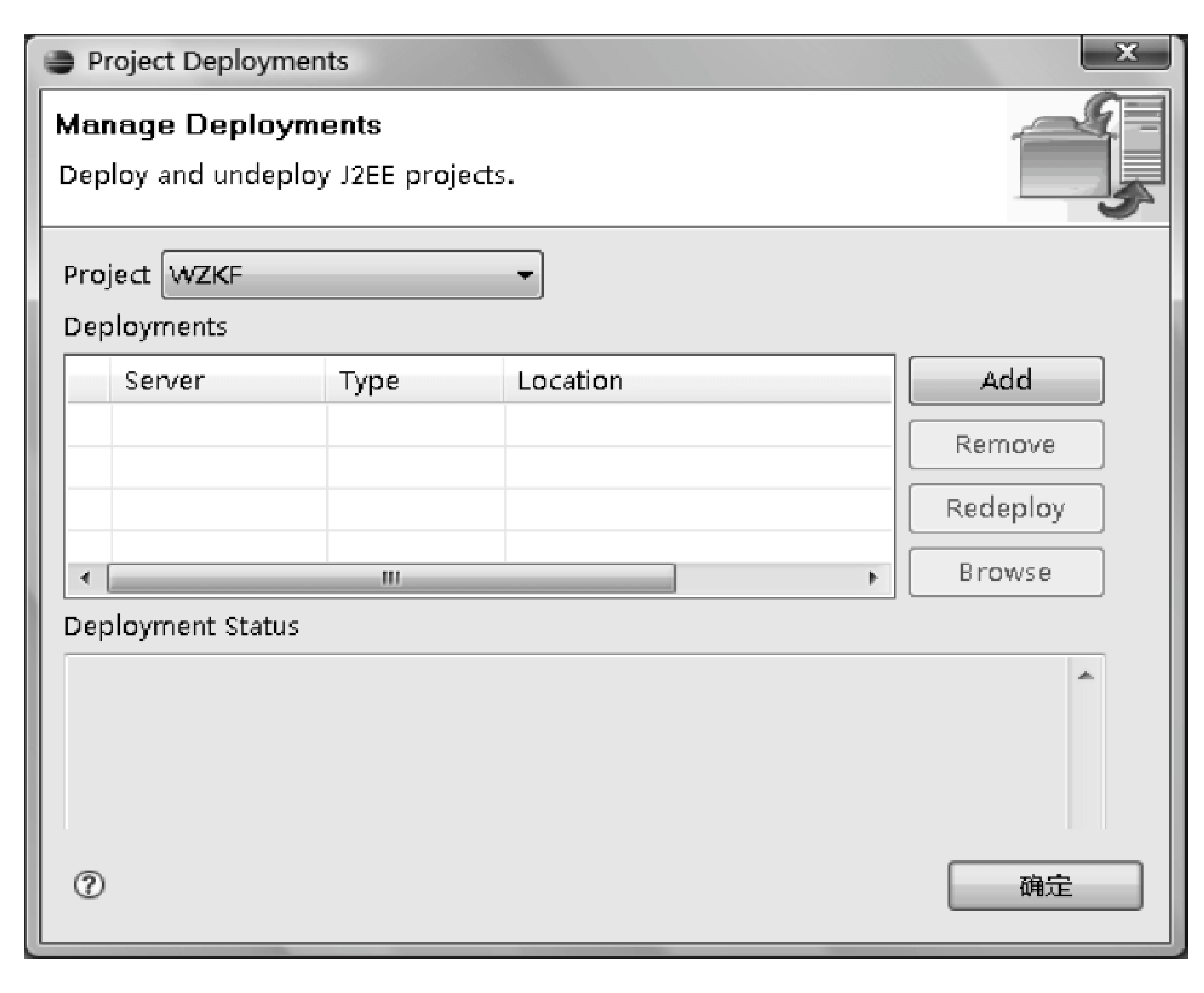


图 8.10　Project Deployments 对话框

(2) 在打开的 New Deployment 新部署对话框中，从 Server 下拉选项中选择 Tomcat 6，在 Deploy type 选项中选择 Exploded Archive，即使用目录方式部属，不打成 war 包，如图 8.11 所示，最后单击“完成”按钮返回上一个界面 Project Deployments 对话框。

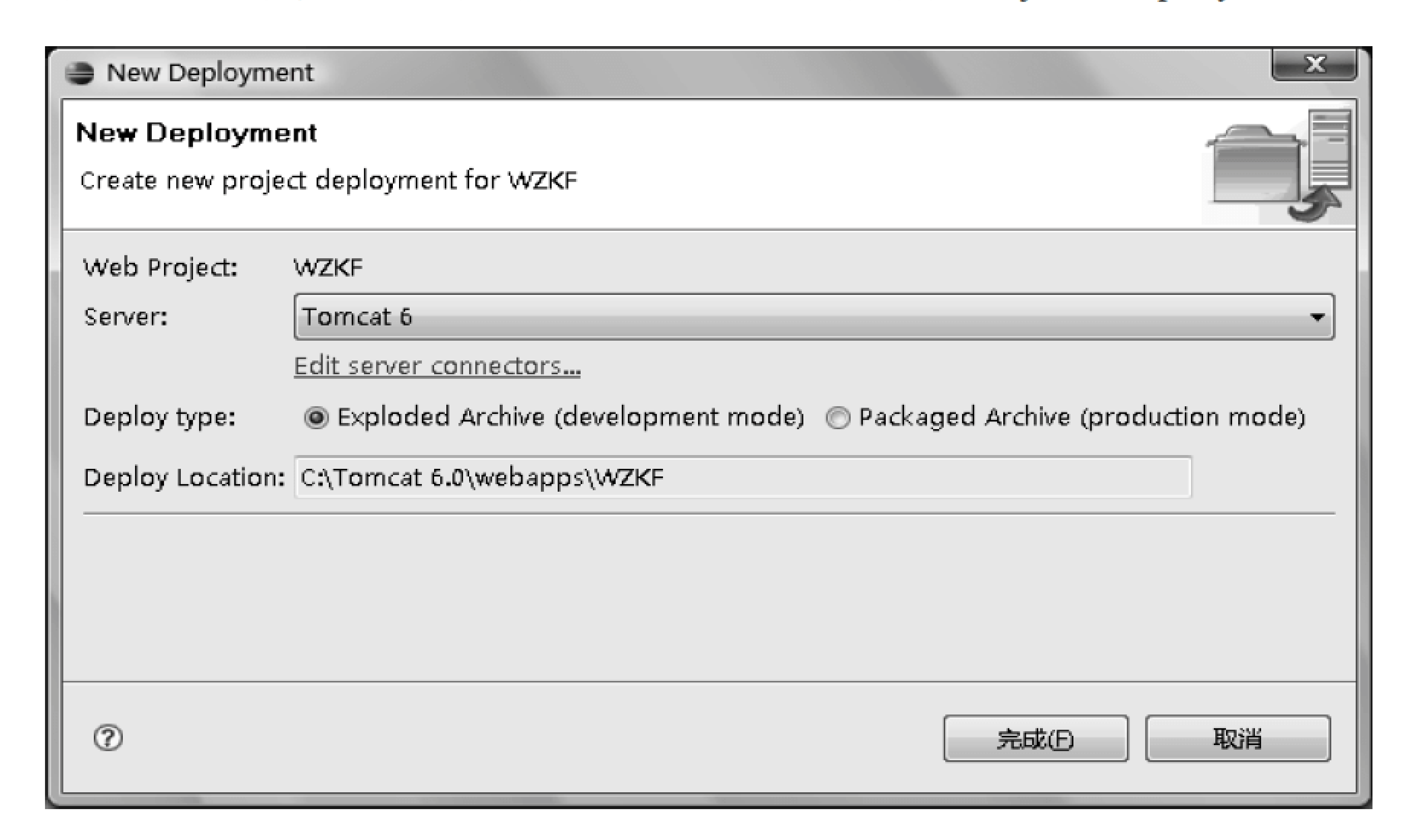


图 8.11　New Deployment(新部署)对话框

(3) 在 Project Deployments 可以看到部署的虚拟目录在 Tomcat 6 服务器目录中对应的物理路径，如图 8.12 所示。此时，原来灰显的按钮变为可用，单击 Remove 和 Redeploy 按钮可以移除部署和重新部署。单击 OK 按钮，即可完成虚拟目录的定义。

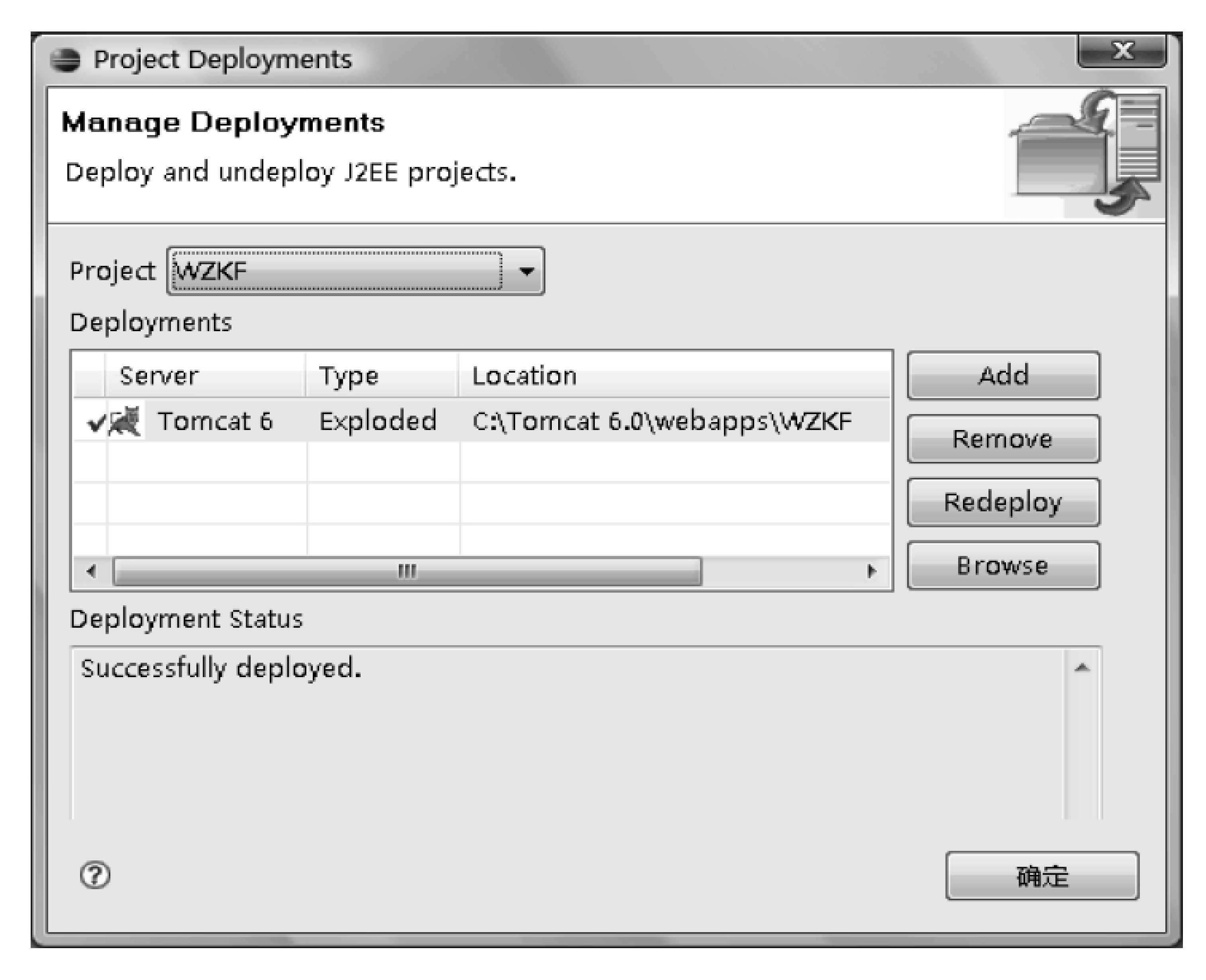


图 8.12　虚拟目录 WZKF 在服务器目录中对应的物理目录

3. 察看 WZKF 工程项目中的内容

单击 WZKF 工程项目右边的“折叠”按钮⊞，可以看到，需要的类都已经自动加载了进

来，还有一个 J2EE 的类环境变量，如图 8.13 所示，这就是使用 MyEclipse 的方便之处。

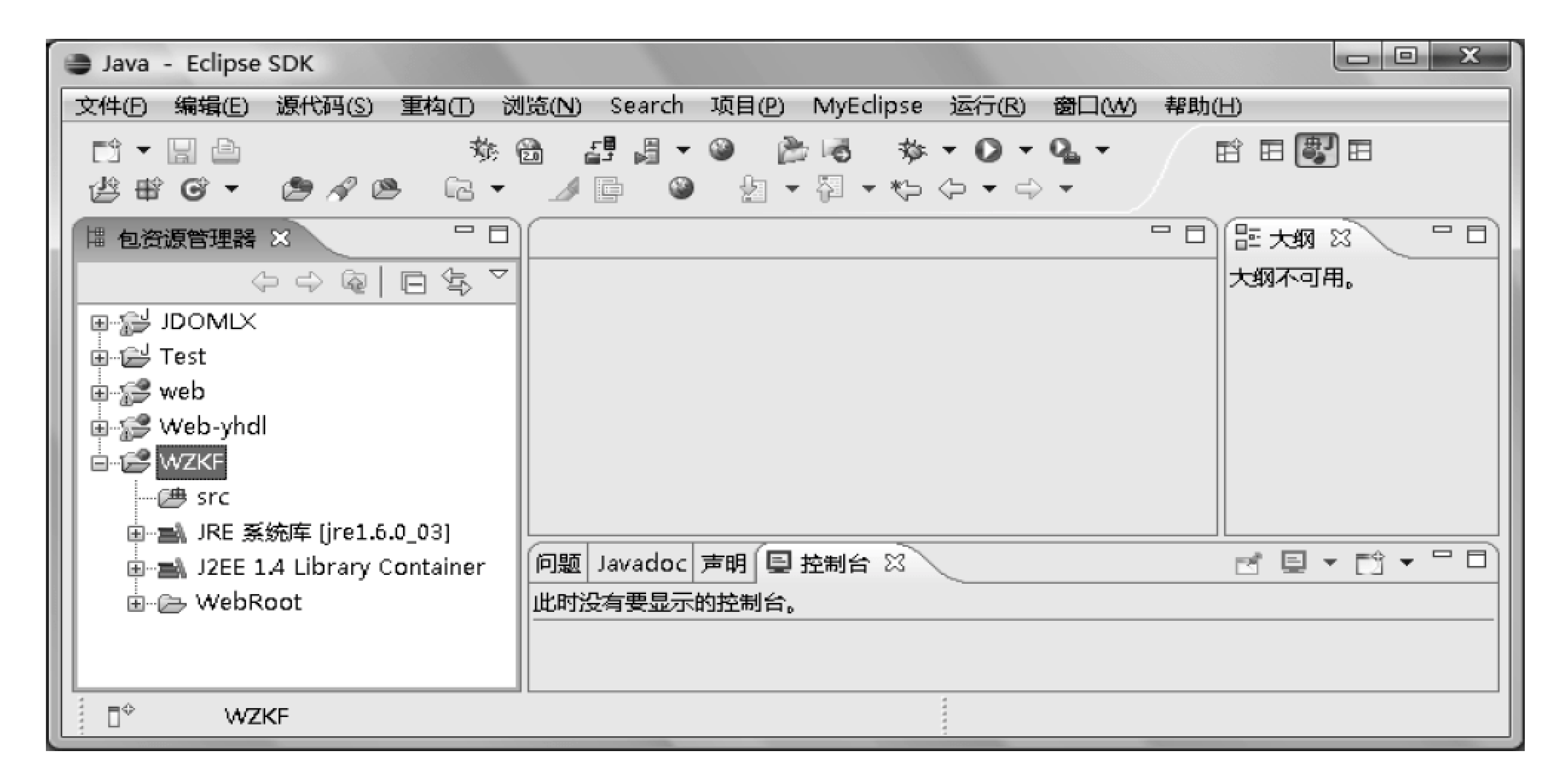


图 8.13　WZKF 项目包含的内容

8.2.4　创建用户登录系统中“视图”模块中的应用程序

1. 创建用户登录的 JSP 页面

(1) 在 MyEclipse 工作界面上右击 WZKF 下的 WebRoot 子目录，在弹出的快捷菜单中选择 New→“文件夹”命令，打开“新建文件夹”对话框，在“文件夹名”文本框输入 yhdl，如图 8.14 所示。然后单击“完成”按钮。即可在 WebRoot 子目录下创建一个子文件夹。



图 8.14　“新建文件夹”对话框

（2）在 MyEclipse 工作界面右击 WZKF/WebRoot/yhdl 子文件夹名，在弹出的快捷菜单中选择 New→“其他”命令，打开 New（新建）对话框，从中选择 Web→JSP 选项，如图 8.15 所示。然后单击“下一步”按钮。

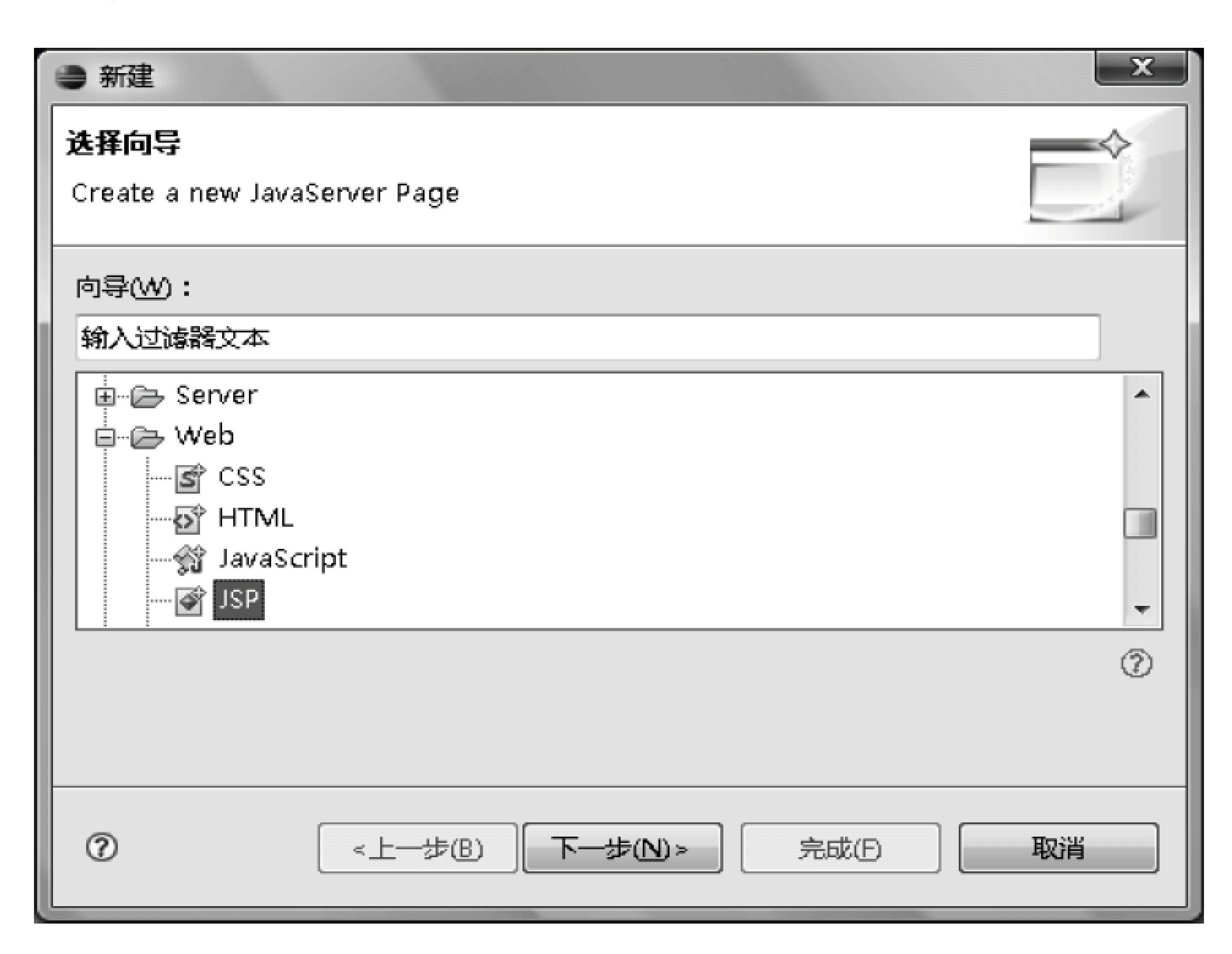


图 8.15 “新建”对话框

（3）在“新建”对话框的“文件名”文本框中输入 login.jsp，如图 8.16 所示。



图 8.16 输入文件名

(4) 在“新建”对话框单击“完成”按钮，将打开 login.jsp 编辑文件视图窗口，如图 8.17 所示。

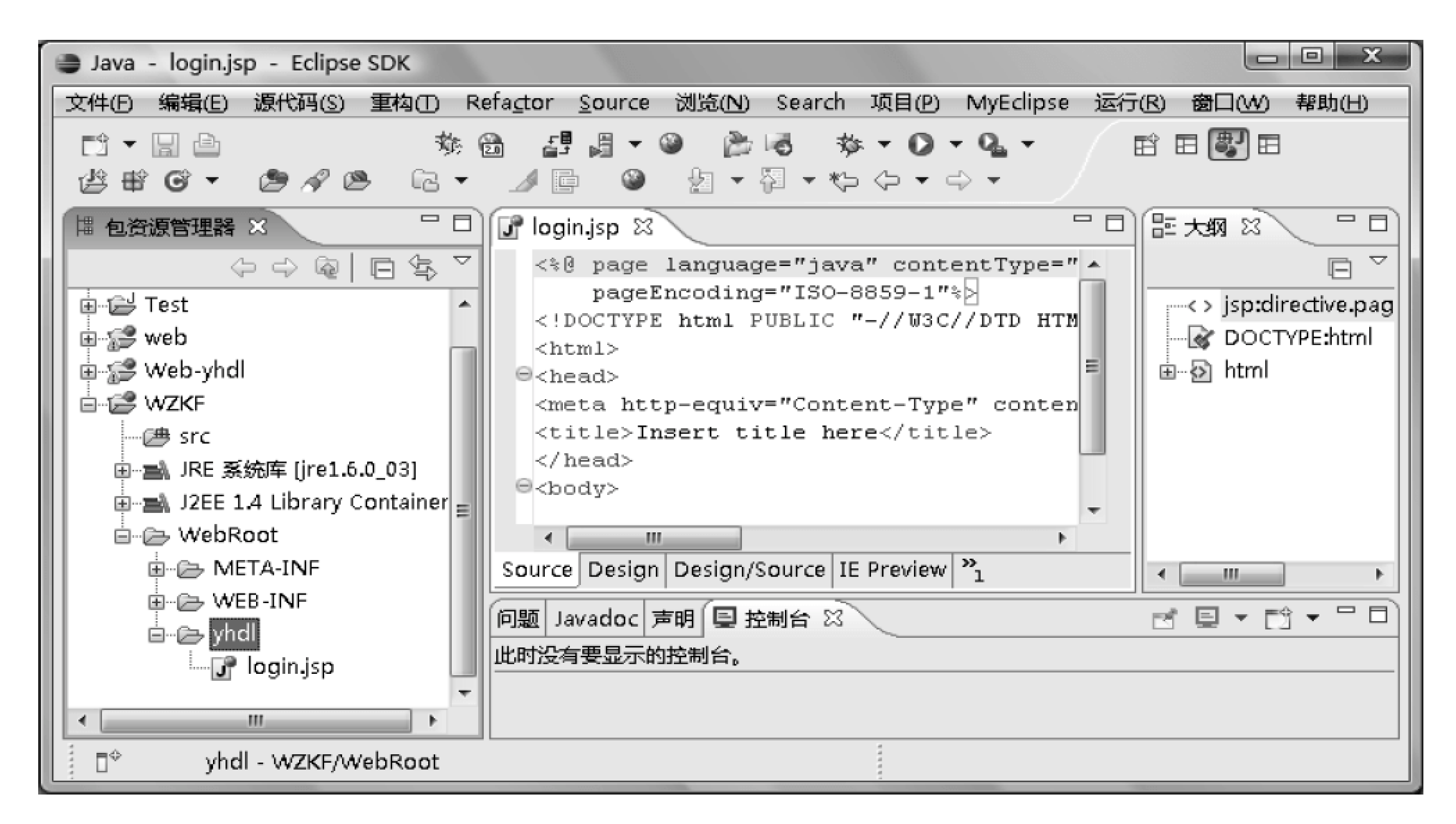


图 8.17　编辑文件视图窗口

在其中输入如下所示的用户登录 login.jsp 文件的源代码：

```
<%@page language="java"pageEncoding="GB2312"%>
<html>
  <head><title>用户登录页面</title></head>
   <body>
  <div align="center">
  <%
   if(!((String)request.getAttribute("error.message")==null)){
  %>
    <form name="regForm" method="post" action="./Check">
    <%}else{%>
    <form name="regForm" method="post" action="../Check">
    <%}%>
    用户名<input name="username" type="text" id="username"><br/>
    密  码<input name="password" type="password" id="password"><br/>
    <input type="Submit" name="submit" value="提交"/>
    <input type="Reset" name="reset" value="重置"/>
    <br/><br/>
    <font color="red">

    <%
    if(!((String)request.getAttribute("error.message")==null)){
        out.print(request.getAttribute("error.message"));
    }
```

```
      %>
      </font>
      </form>
    </div>
  </body>
</html>
```

注意：“./Check”表示为当前目录，“../Check”表示为当前路径的父目录。

(5) 编辑文件视图窗口下方单击 Design 标签，会出现 JSP 文件的设计视图，是一个可视界面，如图 8.18 所示，在这里可以方便地对 JSP 页面进行设置。

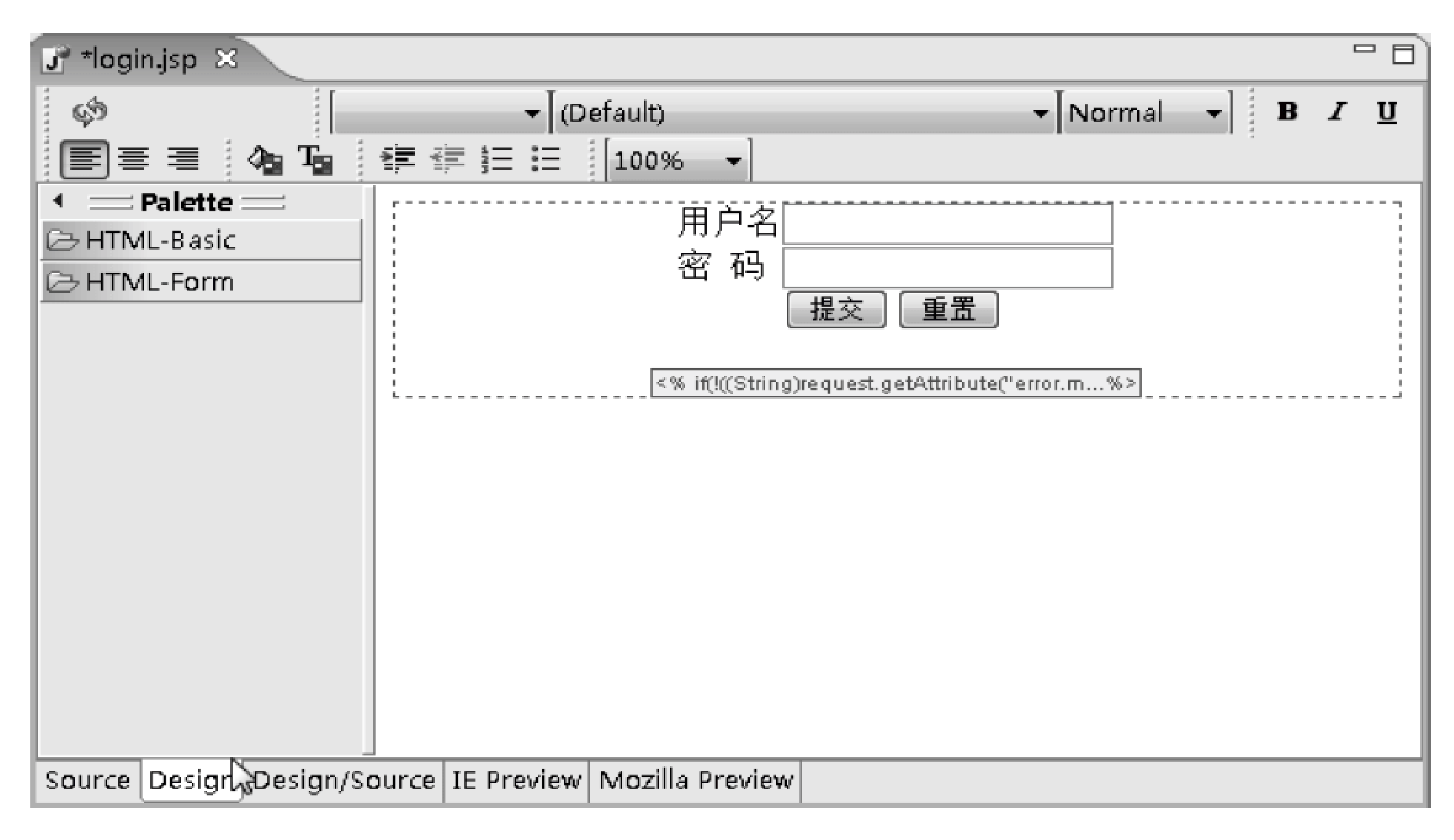


图 8.18 JSP 设计视图窗口

2. 创建用户登录成功的 JSP 页面

在 JSP 编辑视图窗口，输入如下所示的 success.jsp 源代码：

```
<%@page language="java" pageEncoding="GB2312"%>
<html>
  <head>
    <title>用户登录成功页面</title>
  </head>
   <body>
  <div align="center">
    祝贺你<%=(String)request.getAttribute("name")%>登录成功！
   </div>
  </body>
</html>
```

单击菜单栏“文件”→“另存为”菜单命令，可保存生成 success.jsp 文件。

8.2.5 创建用户登录系统中“控制器”模块中的应用程序

Servlet 应用程序是由负责对一个 HTTP 请求产生适当响应的 Java 类构成的，如同创建 JavaBean 一样，创建 Servlet 从编写 Java 源程序开始，其存放位置与 JavaBean 相同。在 MyEclipse 中在 WZKF 项目中自动为存放 Java 源程序提供了一个名为 src 的子目录，可以在其中定义不同的包，存放不同应用使用的用于创建 JavaBean 与 Servlet 的 Java 源程序，运行这些 Java 源程序后，MyEclipse 会自动将其 class 文件存放在 WebRoot\WEB-INF\classes\包名中，这也是使用 MyEclipse 的方便之处。

1. 创建 Servlet 控制器程序 Check.java

(1) 右击 WZKF 工程项目 src 子目录，在弹出的快捷菜单中选择 New→class(类)命令，打开 New Java(新建 Java 类)对话框，如图 8.17 所示。

(2) 在“包”文本框输入 mypack，在“名称”文本框输入 Check，如图 8.19 所示。最后单击“完成”按钮。将在 WZKF/src 子目录下创建一个 mypack 子文件夹，并在其中创建一个自动生成的 Check.java 文件，同时打开 Check.java 文件编辑视图窗口，如图 8.20 所示。



图 8.19 定义构成 Servlet 的 Java 程序名

(3) 在 Check.java 文件编辑视图窗口输入如下所示的 Check.java 源程序：

```
package mypack;

import java.io.* ;
```

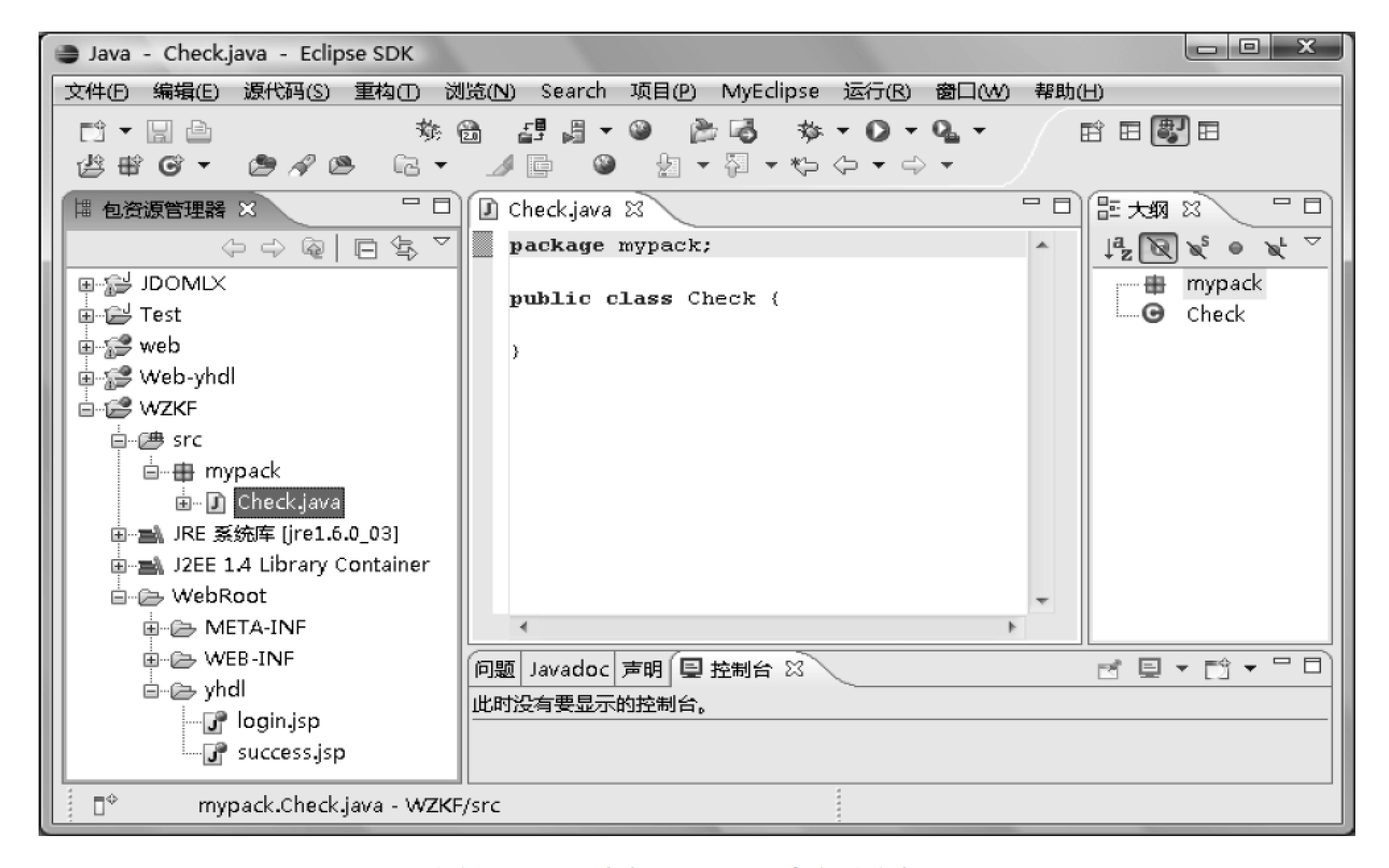


图 8.20 编辑 Java 程序视图窗口

```
import java.sql.Connection;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Check extends HttpServlet {
 private CDB cdb=null;

 String page=null;

 public Check() {super();}

 public void init() {cdb=new CDB();}

 public void destroy() {super.destroy();}

 public void doGet(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
    try{
    String yhm=(String) request.getParameter("username");
```

```
        String mm= (String) request.getParameter("password");
        Connection con= cdb.getCon();

        boolean temp= CD.isValid(con, yhm, mm);
        cdb.closeCon(con);
        if (!temp) {

            request.setAttribute("error.message", "非法登录!");
            page= "yhdl/login.jsp";
            } else{
            request.setAttribute("name", yhm);
            page= "yhdl/success.jsp";

        }

        request.getRequestDispatcher(page).forward(request,response);
        }catch(Exception e){
            e.printStackTrace();
        }
        return;
    }
    }
```

由于还没有编写 JavaBean 的应用程序，这时，Check. java 程序会显示一些错误。要避免这种问题，可等创建“模型”模块中的 JavaBean 后再运行该程序。

运行 Check. java 程序时，单击工具栏上的“运行”按钮，在其快捷下拉菜单中选择“运行方式”→Java Bean。

2. Servlet 的知识点归纳

(1) 创建 Servlet 需要导入的类库

创建 Servlet 的 Java 类时必须导入 java. io. *、javax. servlet. * 和 javax. servlet. http. * 类库。

(2) Servlet 是一个 Java 类

Servlet 是一个 Java 类，但与一般的 Java 类不同的是它不需要 main()方法，这点与 JavaBean 相同。可以将它看成为运行在服务器上的 applet 小程序。

(3) Servlet 是继承自 HttpServlet 类

因为 Servlet 是继承自 HttpServlet 的子类，必须覆盖 doGet()、doPost()、doPut()、doDelete()等方法之一。通常覆盖 doGet()方法，这是 HTTP 默认的请求方法。这些方法用来处理用户的请求与对用户的响应，在方法内，用户请求表示为一个 HttpServletRequest 对象，例如 request；对用户的响应表示为一个 HttpServletResponse 对象，例如 response。

init()方法是初始化方法，例如，在初始化方法中创建打开数据库的连接对象。

destroy()方法用来销毁 Servlet。

3. 修改 web. xml 文件

展开 WebRoot 目录下的 WEB-INF 子目录，双击其中的 web. xml 文件，在文件编辑视图窗口打开该文件，输入如下所示代码，修改 web. xml 文件并保存。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>Check</servlet-name>
    <servlet-class>mypack.Check</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Check</servlet-name>
    <url-pattern>/Check</url-pattern>
  </servlet-mapping>
</web-app>
```

4. web. xml 文件中主要元素含义

1）servlet 元素

servlet 元素用来给 Web 应用中使用的 servlet 或 JSP 程序注册名称，servlet-name 子元素用来定义 servlet 或 JSP 程序的名称；servlet-class 子元素用来定义 servlet 类文件所在的位置。

在 web. xml 中的 servlet 元素表示位于 WEB-INF/classes/mypack/Check 的 servlet 注册名为 Check，注册名可以与原来的 servlet 类名不同。

servlet 的注册名可在 URL 中使用，利于提高安全性。

2）servlet-mapping 元素的作用

servlet-mapping 元素可以定制默认的 URL。servlet-name 子元素用来定义 servlet 的名称；url-pattern 子元素用来定义相对于一个 Web 服务根目录的 URL，url-pattern 元素的值必须以斜杠(/)起始。

在 web. xml 中的 servlet-mapping 元素表示默认的 URL 地址

```
http://localhost:8080/WZKF/WEB-INF/classes/mypack/Check
```

可使用 http://localhost:8080/WZKF/Check 代替，通过注册名与此 URL 可以找到 servlet。

8.2.6 创建用户登录系统中“模型”模块中的应用程序

1. 创建 JavaBean 程序 CDB.java

在“WZKF/src/mypack”中创建用于连接数据源 jsp 的 CDB.java，其源代码如下所示：

```
package mypack;
import java.sql.* ;
public class CDB {
    private Connection con=null;
    public CDB() {      }
    public Connection getCon()   {
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con=DriverManager.getConnection("jdbc:odbc:jsp");
        } catch (Exception e) { System.err.println("数据库连接:"+e.getMessage()); }
        return con;
    }
    public void closeCon(Connection con) {
        try{if(con!=null)con.close();}catch(SQLException ex) {
            System.err.println("释放对象:"+ex.getMessage());
        }
    }
}
```

运行 CDB.java 程序，单击工具栏上的“运行”按钮，在其快捷下拉菜单中选择“运行方式”→Java Bean。

2. 创建 JavaBean 程序 CD.java

在 WZKF/src/mypack 中创建用于判断数据的 CD.java，其源代码如下所示：

```
package mypack;
import java.sql.* ;
public class CD {
    public CD() {      }
    public static boolean isValid(Connection con,String yhm,String mm)   {
        boolean temp= false;
        PreparedStatement ps=null;
        ResultSet rs=null;
        String sql="select admname from admins where admname=?and admpwd=?";
        try {
            //查询记录
            ps=con.prepareStatement(sql);
```

```
                ps.setString(1,yhm);
                ps.setString(2,mm);
                rs= ps.executeQuery();
                if(rs.next())temp=true;
                else temp=false;
                rs.close();
                ps.close();
                } catch (Exception e) {e.printStackTrace();}
            return temp;
        }
    }
```

运行 CD.java 程序。

8.2.7 运行用户登录系统

1. 输入网址打开 login.jsp 登录页面

单击 MyEclipse 工具栏上的 MyEclipse Web Browser 按钮，在地址栏输入 http://localhost:8080/WZKF/yhdl/login.jsp，单击按钮 或按 Enter 键，可看到如图 8.21 所示页面。

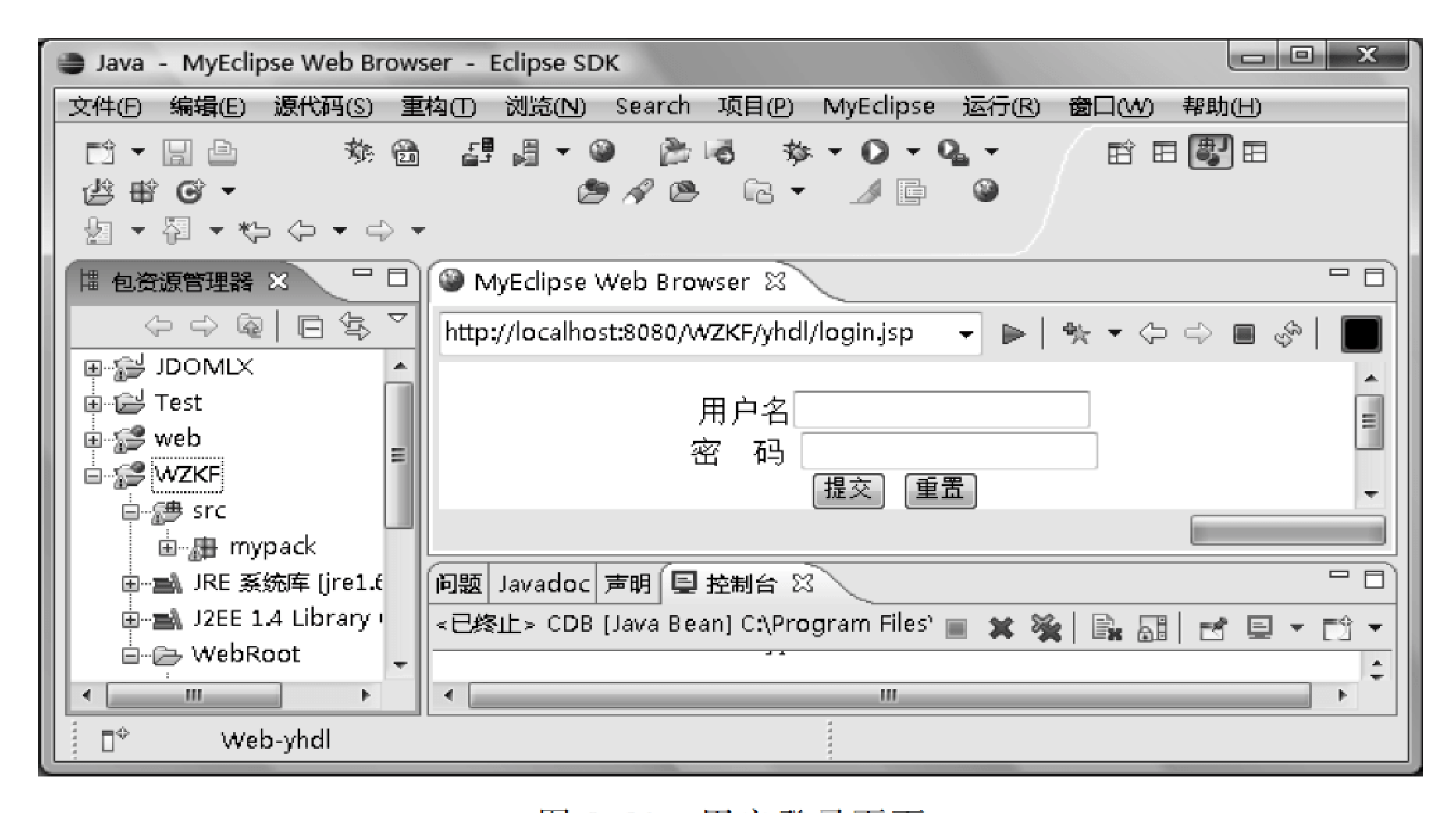


图 8.21 用户登录页面

输入用户名 lpshao，密码××，单击“提交”按钮，如果密码正确，可看到如图 8.22 所示页面。如果密码或用户名不正确，可看到如图 8.23 所示页面。

2. 知识点归纳

在基于 Java 的 Web 应用系统中是通过 JSP、Servlet 与 JavaBean 来实现 MVC 设计模式的。其中，视图就是客户端浏览器中看到的 jsp 页面，它们是负责与用户交互的前台

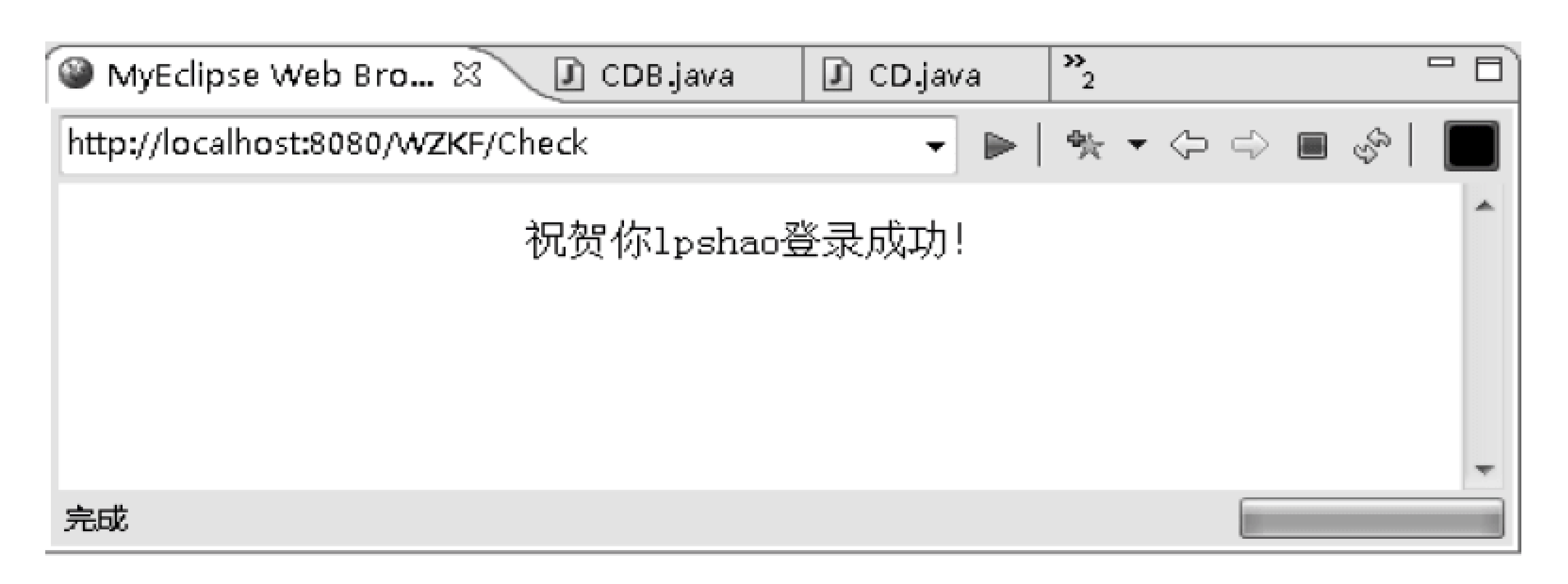


图 8.22　用户成功登录页面



图 8.23　用户登录失败页面

页面，用来接收用户输入数据，或向用户返回数据处理结果。控制器用来解决控制页面流程，转发，提交，重定向、验证处理数据等任务，它将用户输入的数据进行转换交给模型处理，然后根据模型得到的业务处理结果决定使用哪个视图(JSP 页面)展现结果，控制器应用程序主要由 Servlet 构成。模型用来完成业务逻辑，负责访问数据、处理数据或临时存储数据等工作，模型应用程序主要由 JavaBean 构成。

8.3　Ajax 开发模式

8.3.1　什么是 Ajax 开发模式

1. 什么是 Ajax

Ajax 是 Asynchronous JavaScript and XML 的缩写，意思是异步的 JavaScript 与 XML。它可以实现客户端的异步请求操作，这样可以实现在不需要刷新页面的情况下与服务器进行通信，从而减少了用户的等待时间。

2. Ajax 的异步概念

传统 Web 开发模式是一种同步概念。用户必须等待每个请求，当一个请求完成后才能获得结果，在使用完这些结果后才会发出新的请求。例如，当用户请求了一篇文章，他肯定会在阅读完这篇文章后才会去获取其他数据，否则当前文章页面将被刷新，无法阅读。它属于“请求→刷新→响应”模式，用户只有等请求完成后才能进行用户操作，操作完

成后才能提交下一个请求，用户行为和服务器行为是一种同步的关系。

Ajax开发模式是一种异步概念。这意味着客户端和服务器端不必再相互等待，而是进行一种并发的操作。用户再发送请求以后可以继续当前工作，包括浏览或提交信息，在服务器响应完成之后，Ajax引擎会将更新的数据显示给用户看，而用户则根据响应内容来决定自己下一步的行为。

3. Ajax的优点

与传统的Web应用不同，Ajax在用户与服务器之间引入了一个Ajax引擎，消除了网络交互过程中的处理→等待→处理的缺点。Ajax的优点具体表现在以下几方面：

（1）减轻服务器的负担。Ajax的原则是“按需求获取数据”，这可以最大程度地减少冗余请求和响应对服务器造成的负担。

（2）可以把一部分以前由服务器负担的工作转移到客户端，利用客户端闲置的资源进行处理，减轻服务器和带宽的负担，节约空间和成本。

（3）无需刷新更新页面，从而使用户不用像以前一样，在服务器处理数据时只能在死板的白屏签焦急地等待。Ajax使用XMLHttpRequest对象发送请求并得到服务器响应，在不需要重新载入整个页面的情况下，可以通过DOM及时将更新的内容显示在页面上。

（4）可以调用XML等外部数据，进一步促进页面显示和数据的分离。

（5）是基于标准化的并被广泛支持的技术，不需要下载插件或小程序。

4. Ajax的缺点

（1）Ajax通信常常是一个数据片段，例如，xml.html片段，或者纯文本，这些信息不是完整的html文档，而且通过后台来异步获取，不能被搜索引擎发现并缓存。

（2）Ajax将大量的运算从服务器端转移到了客户端，意味着浏览器将承受更大的负担，不再是简单的文档显示。Ajax中的核心语言是JavaScript，作为一门解释型的脚本语言，其运行效率不是很高。而且依赖不同的浏览器实现。

8.3.2 Ajax应用实例——用户注册页面

1. 新用户注册页面文件

新用户注册页面如图8.24所示。该页面的特点是可以在输入用户名后立即检测数据库中是否存在该名称，可以根据选择的不同省名显示该省的市名。还可以立即判断是否填写了必填写的数据。

新用户注册页面文件为index.jsp，源代码如下：

```
<%@page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
<html>
<head>
```

图 8.24　新用户注册页面

```
<title>新用户注册页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script src="function.js" language="javascript">
</script>
<style type="text/css">
.STYLE4 {font-size: 14px; font-weight: bold;}
</style>
</head>
<%@include file="jdbc-access.jsp"%>
<%
String sql="select ID,省 from 省";
ResultSet rs=s.executeQuery(sql);
%>
新用户注册页面
<form name="form1" method="post" action="">
    <table width="76%" height="295" border="0" cellpadding="0" cellspacing="0">
    <tr><td width="27%" align="center"><span class="STYLE4">用户名:</span></td>
                <td colspan="4"><input name="username" type="text" class="table" id="username">
                        <a href="#" onClick="checkName()">[检测用户名]</a> *</td>
    </tr>
    <tr><td align="center"><span class="STYLE4">密码:</span></td>
        <td colspan="4" class="word"><input name="PWD1" type="password" class="table" id="PWD1">
                      *</td>
```

```
  </tr>
  <tr><td align="center"><span class="STYLE4">确认密码：</span></td>
     <td colspan="4" class="word"><input name="PWD2" type="password" class="table"
id="PWD2">
     * </td>
 </tr>
 <tr><td align="center"><span class="STYLE4">生日：</span></td>
   <td colspan="4" class="word"> <input name="birthday" type="text" class=
"table" id="birthday">
                    (格式：1980-07-17)</td>
 </tr>
 <tr><td align="center"><span class="STYLE4">性别：</span></td>
   <td colspan="4"><input name="sex" type="radio" value="男" checked>男生
                    <input type="radio" name="sex" value="女">女生</td>
</tr>
<tr><td align="center" class="STYLE4">省：</td>
   <td width="27%"><select name="supertype" id="supertype" onChange=
"f1(this.value)">
<%
   while(rs.next())
       {
%>
    <option value="<%=rs.getInt(1)%>"><%=rs.getString(2)%></option>
                    <%}%>
                  </select></td>
   <td width="22%" align="center"><span class="STYLE4">市：</span></td>
   <td width="24%" id="subType"> </td>
</tr>
<tr><td align="center"><span class="STYLE4">OICQ 号码：</span></td>
  <td colspan="4"><input name="OICQ" type="text" class="table" id="OICQ"></td>
</tr>
<tr><td align="center"><span class="STYLE4">个人主页：</span></td>
   <td colspan="4"> <input name="homepage" type="text" class="table" id="homepage"
size="50"></td>
</tr>
<tr><td align="center"><span class="STYLE4">地址：</span></td>
  <td colspan="4"><input name="address" type="text" class="table" id="address" size=
"50"></td>
</tr>
   <tr><td align="center"><span class="STYLE4">邮政编码：</span></td>
    <td colspan="4"><input name="postcode" type="text" class="table" id="postcode">
</td>
 </tr>
<tr><td align="center"><span class="STYLE4">Email:</span></td>
```

```
    <td colspan="4" class="word"><input name="Email" type="text" class="table" id="
Email" size="46">
                    * </td>
</tr>
<tr><td colspan="5" align="center">
     <input name="save" type="button" class="btn_grey" onClick="mycheck()"value="保存"
>
                     
     <input name="re" type="reset" class="btn_grey" value="重置"></td>
</tr>
</table>
</form>
</body>
</html>
```

2. 数据库连接文件

为了保持用户信息，可先在数据库中建立一个表，在新用户注册页面文件 index. jsp 中使用的数据库连接文件 jdbc-access. jsp，源代码如下：

```
<%@page import="java.sql.*"%>
<%Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String url="jdbc:odbc:driver={Microsoft Access Driver (*.mdb)};DBQ=D:/ERPSS/shu/DBA.
mdb";
Connection c=DriverManager.getConnection(url,"","");
Statement
s=c.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
%>
```

可见，数据库文件 DBA. mdb 存放在 D:/ERPSS/shu 文件夹下。

3. 用户名检测文件

在新用户注册页面文件 index. jsp 中要根据新用户输入的用户名检测数据库中是否存在，需要调用函数 checkName()，而该函数需要使用 checkName. jsp 文件与数据库进行连接与判别。checkName. jsp 源代码如下：

```
<%@page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
<%@include file="jdbc-access.jsp"%>
<%
request.setCharacterEncoding("GB2312");
String username=request.getParameter("username");
ResultSet rs=s.executeQuery("select * from 客户 where 用户名='"+username+"'");
if (rs.next()){
    out.println("很抱歉!用户名["+username+"]已经被注册!");
}else{
```

```
out.println("祝贺您!用户名["+username+"]没有被注册!");
}
%>
```

4. 根据选择的省名产生不同市名的动态级联菜单文件

在新用户注册页面文件 index.jsp 中要动态产生级联菜单,其省名与市名可以由数据库中的数据产生需要使用函数 f1(this.value),而该函数需要使用 select.jsp 文件连接数据库,根据指定的省确定有哪些市,显示结果如图 8.25 所示。

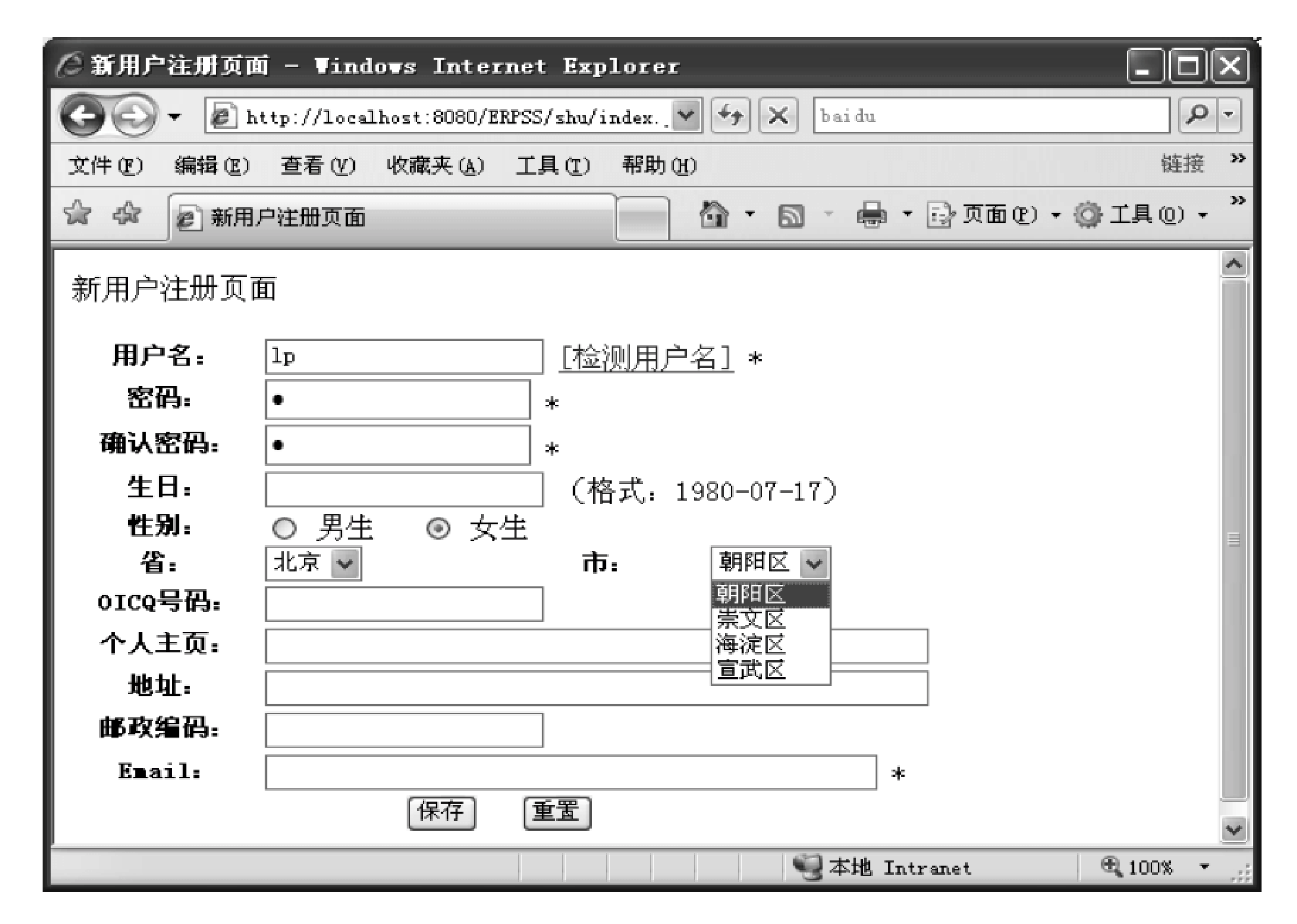


图 8.25 动态产生的级联菜单

select.jsp 文件源代码如下:

```
<%@page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
<%@include file="jdbc-access.jsp"%>
<select name="typeID" id="typeID">
<%
if((request.getParameter("ID")==null)||(request.getParameter("ID")==""))
{
%>
<option value="0">无选择</option>
<%
}
else
{
int ID=Integer.parseInt(request.getParameter("ID"));
```

```
ResultSet rs=s.executeQuery("select * from 市 where 省="+ID);
 while(rs.next())
  {
%>
<option value="<%=rs.getInt(1)%>"><%=rs.getString(2)%></option>
<%}
}%>
 </select>
```

5. 保持多个函数的 JS 文件

新用户注册页面文件 index.jsp 中要使用多个函数，可创建一个 JS 文件来保存它们，还可以在其他地方使用这些函数，函数文件 function.js 的源代码如下：

```
function checkName() {

    var username=form1.username.value;
    if(username=="") {
        window.alert("请填写用户名!");
        form1.username.focus();
        return false;
    }
    else {
        url="checkName.jsp?username="+username;
        send_request(url);
    }
}
function mycheck(){                                   //判别是否为空或输入不一致
              if(form1.username.value==""){
                alert("请输入用户名!");form1.username.focus();return; }
              if(form1.PWD1.value==""){
                alert("请输入密码!");form1.PWD1.focus();return;}
              if(form1.PWD2.value==""){
                alert("请确认密码!");form1.PWD2.focus();return;}
              if(form1.PWD1.value!=form1.PWD2.value){
                alert("您两次输入的密码不一致,请重新输入!");
                form1.PWD1.value="";form1.PWD2.value="";form1.PWD1.focus();return;}
            if(form1.Email.value==""){
                alert("请输入 Email 地址!");form1.Email.focus();return; }
            form1.submit();
          }
 function f1(ID){
    send_request2("select.jsp?ID="+ID);             //实现级联下拉列表
}
```

```
//用户名检测函数
var http_request=false;
function send_request(url){
            //初始化、指定处理函数、发送请求的函数
            http_request=false;
            //开始初始化 XMLHttpRequest 对象
            if(window.XMLHttpRequest){
                //Mozilla 浏览器
                http_request=new XMLHttpRequest();
                if(http_request.overrideMimeType){
                    //设置 MiME 类别
                    http_request.overrideMimeType("text/xml");
                }
            } else if(window.ActiveXObject){
                //IE 浏览器
                try{
                    http_request=new ActiveXObject("Msxml2.XMLHTTP");
                }catch(e){
                    try{
                        http_request=new ActiveXObject("Microsoft.XMLHTTP");
                    }catch(e){}
                }
            }
            if(!http_request){
                //异常,创建对象实例失败
                window.alert("不能创建 XMLHttpRequest 对象实例");
                return false;
            }
            http_request.onreadystatechange=processRequest;
            //确定发送请求的方式和 url 以及是否同步执行下段代码

            http_request.open("GET",url,true);

            http_request.send(null);
    }
    //处理返回信息的函数
    function processRequest(){
            if(http_request.readyState==4){
                //判断对象状态
                    //信息已经成功返回,开始处理信息
                if(http_request.status==200){
                      alert(http_request.responseText);

                }else{
```

```
                alert('请求的界面异常!');
                alert(http_request.responseText);
            }
        }
    }
    //级联下拉列表*******************************************************************
function send_request2(url){
            //初始化、指定处理函数、发送请求的函数
            http_request=false;
            //开始初始化 XMLHttpRequest 对象
            if(window.XMLHttpRequest){
                //Mozilla 浏览器
                http_request=new XMLHttpRequest();
                if(http_request.overrideMimeType){
                    //设置 MiME 类别
                    http_request.overrideMimeType("text/xml");
                }
            } else if(window.ActiveXObject){
                //IE 浏览器
                try{
                    http_request=new ActiveXObject("Msxml2.XMLHTTP");
                }catch(e){
                    try{
                        http_request=new ActiveXObject("Microsoft.XMLHTTP");
                    }catch(e){}
                }
            }
            if(!http_request){
                //异常,创建对象实例失败
                window.alert("不能创建 XMLHttpRequest 对象实例");
                return false;
            }
            http_request.onreadystatechange=processRequest2;
            //确定发送请求的方式和 url 以及是否同步执行下段代码

            http_request.open("GET",url,true);

            http_request.send(null);
    }
    //处理返回信息的函数
    function processRequest2(){
            if(http_request.readyState==4){
                //判断对象状态
                    //信息已经成功返回,开始处理信息
```

```
                if(http_request.status==200){
                    subType.innerHTML=http_request.responseText;

                }else{
                    alert('请求的界面异常!');
                    alert(http_request.responseText);
                }
            }
        }
```

8.4 Web 开发框架

在 Java Web 应用程序开发时，目前流行许多 Web 开发框架。使用开发框架可以保证快速开发出具有良好架构、易于扩展、易于管理维护的 Web 应用程序。使用框架的目的是使混乱的东西变得结构化。Web 开发框架就是鼓励开发人员使用一系列框架所提供的基础类和类库，避免 JSP 造成的混乱。使用框架不是因为它能让你做什么，而是因为它限制你做什么。

本节的内容简单介绍几种常见的 Web 开发框架，为继续学习 Web 开发技术指明一个方向。

8.4.1 Struts

Struts 是 Apache 软件组织提供的一项开放源码项目，它是一个基于 Sun J2EE 平台的 MVC 框架，主要采用 Servlet 和 JSP 技术来实现，尤其适用于开发大型可扩展的 Web 应用。Struts 具有很高的可配置性，一个前端控制组件、一系列动作类、动作映射、处理 XML 的实用工具类以及服务器端 JavaBeans 的自动填充、支持验证的 Web 表单、国际化支持、生成 HTML、表现逻辑和模版组成了 Struts 的灵魂。由于 Struts 能充分满足应用开发的需求，简单易用，敏捷迅速，受到大家的关注。Struts 把 Servlet、JSP、自定义标签和信息资源(message resources)整合到一个统一的框架中，开发人员利用其进行开发时不用再自己编码实现全套 MVC 模式，极大地节省了时间，所以说 Struts 是一个非常不错的应用框架。

Struts 这个名字来源于在建筑和旧式飞机中使用的支持金属架。Struts 为 Web 应用提供了一个通用的框架，使得开发人员可以把精力集中在如何解决实际业务问题上。此外，Struts 框架提供了许多供扩展和定制的地方，应用程序可以方便的扩展框架，来更好的适应用户的实际需求。

8.4.2 WebWork

WebWork 是一个开放源代码项目，目标是为在短时间内构建复杂网站提供易于理解和维护的工具及开发框架。WebWork 是一个强大的基于 Web 的 MVC 框架，它构建在一个命令模式框架 XWork 之上。

WebWork 有如下特性：

(1) 具有用于处理或代理请求的分发器(Dispatcher)。

(2) 能够支持多种视图技术(JSP、Velocity、JasperReports、XML、FreeMarker)的结果类型。

(3) 拥有一个小而强大的 JSP 标签库和 Velocity 宏定义。

(4) 分发器调用 XWork 活动来访问和操作模型(Model)并为视图显示模型数据提供方便的途径。

WebWork 真正的优势在于它强调简洁和协作能力的根本理念，使用 WebWork 将有助于最小化代码，使开发人员更多的关注业务逻辑(business logic)和建模(modeling)，而不是编写 Servlet 之类的事情。

8.4.3 Spring

Spring 是一个开源框架，它由 Rod Johnson 创建。它是为了解决企业应用开发的复杂性而创建的。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而，Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何 Java 应用都可以从 Spring 中受益。

Spring 之所以存在，是希望 J2EE 更加易用，可用更简单的方法进行 J2EE 开发，同时又不牺牲平台的性能。Spring 在其核心部分提供了一个易于配置的、XML 驱动的反向控制(IoC)容器。IoC 基于所谓的"好莱坞原则"——不要给我们打电话，我们会找你。在这种方案中，应用程序中 Java 对象之间的关系是由容器注入的，而不是通过直接编程实现的。注入分两种形式——构造函数注入和 setter 注入，这取决于容器是通过其构造函数还是通过 mutator 方法把信息注入一个已创建的 Java 对象。

Spring 可以将简单的组件配置、组合成为复杂的应用。在 Spring 中，应用对象，例如注入属性或引用其他 bean，都是通过一个 XML 文件进行配置的声明式地进行组合，这使得配置变得非常直接。

Spring 提供了很多基础功能(事务管理、持久化框架集成等)，Spring 与一个 AOP 框架耦合在一起，而该框架支持非侵入性地添加诸如事务和安全这样的属性，这意味着开发人员可以集中精力创建用于解决业务问题的解决方案，而不会为 J2EE 开发或配置的复杂性而头痛。因为该容器是非侵入性的，所以开发人员不必担心业务代码会被特定于供给商(此处也包括 Spring)的工件所污染。

8.4.4 JVC

JVC 是英文 Java View Control 的简称，中文含义为 Java 视图-控制开发框架。JVC 的目标是构建一个快速开发、可维护的、稳定的 Web 开发框架，通过开放源代码，互相促进开发技术。JVC 参考了目前比较流行的 WebWork、Struts、Spring 的 Web 开发框架的思想，在开发过程中简化了的视图-控制框架，弱化了模型层，弱化模型层的目的是减少模型中大部分的重复工作。

JVC 主要有以下特点：

1）开发简单快速

大部分开发不需要写 JavaBean，系统自带的 Action 能满足 90%需求，只需要编写 page 文件（控制文件）与视图 jsp。

2）提供了 Web 系统开发过程中都需要开发的通用用户权限管理

可以控制角色对每个 Command 的访问权。

3）提供功能强大的验证框架

使用表达式编写验证公式，支持所有逻辑表达式，并且前台 JSP（生成 javascript 脚本）和后台 Servlet（提交到后台的 request 参数）均可以调用。

4）可维护性

一般修改了数据库表需要同时修改 M、V，需要重新编译，虽然简单，但带来了繁琐的操作，而且有可能引起其他错误，而 JVC 框架弱化了 M，不需要编写对象关系映射的类，而是用命名规范使得对于大部分的修改只是需要修改 V（JSP 页面）。

5）自定义 Action 可独立调试

一般 Action 由于需要从 request 得到参数而无法进行独立调试，而 JVC 框架借鉴了 WebWork 方法把 HTTP 的请求封装为 ActionContent 类，除了交互和使用上的方便外，实现了往往开发 Web 比较困难的独立调试。并且为下一步升级到 Web＋EJB 或其他分布式架构打下了基础。

Web 开发框架有很多，有兴趣的读者可以在实践中去发现与了解它们。

8.5 思考与练习

8.5.1 思考题

8-1 MVC 三个英文字母的含义是什么？

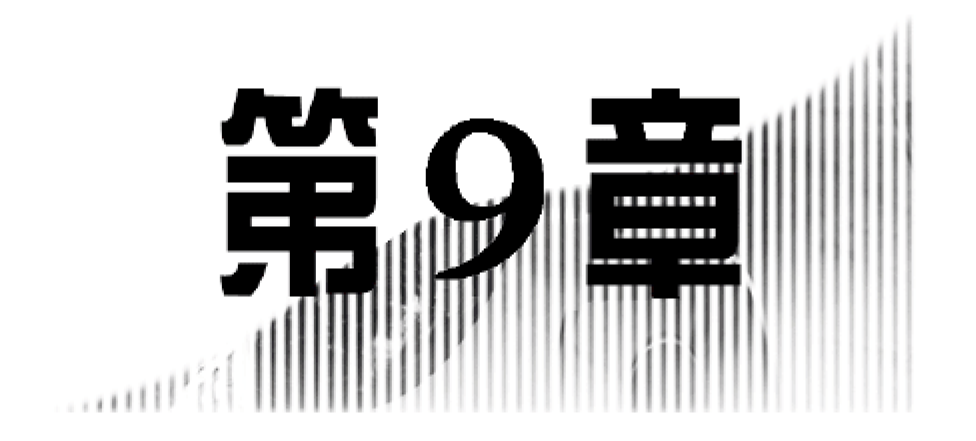
8-2 什么是 MVC 的基本原则？

8-3 MVC 有什么作用？

8-4 有哪些 Web 开发框架？

8.5.2 上机练习

使用 MVC 模式编写一个信息发布、展示、查询与删除的应用程序。



第9章 网站开发各阶段的任务

前面的各章介绍了网站编程的基本技术。要开发网站除了掌握一些网站编程的技术外，还要知道如何进行需求分析、网站规划、网站系统设计等网站开发前期的准备工作，因为网站开发是一项大工程，涉及多个方面，只有做好了各种前期的规划、分析、设计、分工等工作，才能快速、正确的开发一个优秀的网站。本章将以一个"网上体育用品商店"网站的开发作为实例。

本章主要解决以下问题：

- 网站开发有哪些阶段；
- 为什么要进行网站需求调查与分析；
- 为什么要进行网站规划与网站设计；
- 怎样编写网站需求调查与分析报告；
- 怎样编写网站开发规划书；
- 怎样编写网站设计说明书。

9.1 网站开发阶段

本节主要介绍网站开发过程的各个阶段及任务。

开发一个网站，大致可以分为网站开发项目立项、网站需求调查与分析、网站规划、网站设计、网站制作、网站宣传推广、网站运行维护等阶段。

(1) 网站开发项目立项阶段：主要任务是确定开发项目。

(2) 网站需求调查与分析阶段：主要任务是进行网站需求调查与分析，编写网站需求调查与分析报告。

(3) 网站规划阶段：主要任务是进行网站定位，确定网站的目的和功能，编写网站开发规划书。

(4) 网站设计阶段：主要任务是进行网站内容设计、网站结构设计等，编写网站设计说明书。

(5) 网站制作实现阶段：主要任务是用 HTML、JSP、ASP 等网站开发技术制作网页。

(6) 网站宣传推广阶段：主要任务是向亲朋好友推荐，在搜索引擎登记，在新闻组和BBS上作宣传等。

(7) 网站运行维护阶段：主要任务是测试网络，网页的日常维护，网站的宣传推广，网站的优化与更新，网站的备份与恢复。

下面将开发背景限定在用户和开发人员两类人员上，主要介绍项目立项、需求调查与分析、网站规划、网站设计阶段的内容。读者可充当这两种角色。

9.2 项目立项阶段

本节主要介绍项目立项阶段的主要任务——确定开发项目，即为谁开发一个什么类型的网站。

现在越来越多的企业、单位、组织要开发功能强大的网站或浏览器/服务器模式的管理信息系统。所以，会有越来越多的用户有开发网站的需求。当用户有了开发网站的需求，经过用户与开发人员双方不断的接洽和了解，并通过基本的可行性讨论，可初步达成开发网站协议，这时就需要将项目立项，签订项目立项书。

接着要成立一个专门的开发网站项目小组，小组成员包括项目经理，网页设计员，编程员，测试员，编辑/文档员等。一般项目实行项目经理制。

在学生学习过程中可以成立开发网站项目小组，以团队的力量共同来开发一个网站。

9.3 网站需求调查与分析阶段

本节的内容主要介绍网站需求调查分析阶段的主要任务，要清楚开发一个具有什么内容、什么功能的网站，即网站都有什么、都能干什么。具体工作任务如下所述。

9.3.1 明确网站开发的目的

在确定用户要开发一个网站后，开发人员首先要明确用户为什么要建立这个网站？是为了给顾客提供服务，还是要在网上宣传自己的公司、产品，是在网上介绍知识、以网会友，还是要给自己的企业建立一个电子商务网站，只有给这些问题一个满意的答案，才能说明明确了建站的目的，下面的工作才能有一个明确的方向。

9.3.2 网站需求调查与分析

当明确网站开发的目的后，开发人员接着要确定网站包括的主要内容和功能，通过网站的内容和功能实现网站开发的目的。网站的内容是否全面、是否符合顾客的需要，网站的功能是否完善、是否满足用户的使用要求，直接决定了网站开发的成败。网站应该包括哪些内容和功能呢？这个问题可以在网站需求分析阶段来解决。

网站需求分析阶段就是专门解决网站的需求，网站需求从开发人员的角度来划分可以分为内容需求和网站的功能需求；从用户角度来考虑可以分为用户实际需求和用户发展需求。

一个网站项目的确立是建立在各种各样的需求上面的，这种需求往往来自于用户的实际需求或者是出于公司自身发展的需要，其中用户的实际需求占了绝大部分。面对使用网站不同层面的用户，开发人员对用户需求的理解程度，在很大程度上决定了网站开发项目的成败。因此如何更好地的了解、分析、明确用户需求，并能使用准确、清晰的文档形式表达出用户需求，使参与项目开发的每个成员在开发过程中能够按照满足用户需求为目的的正确方向进行各种工作，是需求分析阶段解决的主要问题。

要解决需求分析阶段的问题，先要进行调查工作。调查工作可分为以下两种类型：

1. 用户需求调查

为了更好地进行需求分析，开发人员首先要进行用户需求调查。在调查中要做到以下几点：

1）让用户畅所欲言，罗列出所有的需求

让用户将所有的想法尽可能的阐述清楚，并把所有的要求罗列出来，不要遗漏。这时候开发人员不要害怕“勾引”出客户的潜在需求而增加网站开发的工作量，直接明白的把用户的问题和要求一条条地列出来，将用户最原始、最完整的需求准确地记录下来就完成了调查的第一步工作。这个调查工作要多次进行，并要调查各种类型的用户。可在调查前编写调查表。

2）透过现象分析用户潜在的需求

用户往往对需求的概念是非常模糊的，提出的需求一般分不清是内容需求还是功能需求、是实际需求还是发展需求，大多时候给出的需求都是笼统而且尺度难以控制的，这就要求开发人员在倾听了用户的详细说明以后，进行整理和分析，同时预测用户在开发过程中变更及今后应用中可能进行修改升级的潜在需求。

所以用户需求调查的第二步就是开发人员要将用户描述的需求进行分析、归纳和整理，特别要分析整理出用户谈的不多却是技术上实现难度和强度很高的地方。

例如，在为用户设计办公自动化系统的时候，也许就要为用户预留将来与他们的业务单位进行交互的通道；在设计邮件系统的时候要考虑可能会需要广告管理服务器；设计网络电子商店时可以考虑今后增加库存产品进销存统计分析的功能等等；限于时间财力的考虑，客户通常能够接受分阶段实施的开发过程。在进行用户需求分析时，提早为用户设想到今后的需求变更除了使项目开发更加顺利以外，也为今后业务的进一步深入打下了更好的基础。

3）编写用户需求调查报告

用户需求调查的最后一步工作是编写一个用户需求调查报告。主要内容包括进行了什么样调查、调查规模、调查了哪些用户、调查的覆盖率、不同用户的需求（用户提出的需求和经过开发人员不断引导和分析挖掘出的潜在需求）。

用户需求调查报告需要让用户阅读通过并签字认可。把好编写用户需求调查报告这一关，可以杜绝很多因为需求不明或理解偏差造成的失误和项目失败。

需求调查报告的撰写，要尽可能用自然的语言进行描述，因为用户的水平和资历可能有所不同，所以要使用通俗易懂的自然语言来描述，使项目开发人员和用户都能清楚地理解网站需求的含义，不至于在理解上产生偏差。对用户而言，这样的需求描述最接近真实，容易参与修订，并能以此为测试和验收的依据。

比较以下两份关于需求的描述：

(1) 用户在访问主页的时候单击"会员天地"按钮，弹出填写"用户名"和"密码"的页面，输入正确的"用户名"和"密码"后将在新页面中打开"会员天地"的首页，在该页显示所有会员可以使用的功能导航条和最新的导读新闻链接列表。

(2) 站点分为公开和加密两种状态，通过身份验证机制使特有的用户可以访问到加密信息，并提供不同于普通用户的功能。

前段描述可以很容易想象出制作完成后网站是什么样子，而后一段的描述可能会做出无数不同的版本，造成对需求理解的歧义。

2. 对类似网站进行调查

开发人员还要对类似网站进行调查，目的是了解目前类似网站的情况，了解当前类似网站的内容和功能，以拓宽思路，使功能更完善，设计更合理。根据调查结果，结合要开发的网站的特点，通过仔细讨论与分析，给要开发的网站提出一些新的功能需求和内容需求。

3. 类似网站调查报告

对类似网站的调查也要编写一份网站调查报告，主要内容包括：

1）调查规模

调查了哪些网站、调查的覆盖率。

2）调查的内容

被调查网站所具有的内容特色、功能特点。

3）调查的收获

根据调查情况提出开发网站的内容参考建议和网站功能的参考建议。

9.3.3 编写需求分析报告

在用户需求调查和类似网站调查报告的基础上，开发人员要经过多次讨论和认真分析研究编写一份网站需求分析报告。

1. 网站需求分析报告主要内容

1）确定网站的用户角色

即哪些人员浏览和使用网站。如访客、会员、重要客户、前台管理员、网站管理员、业务员等。

2）分析用户需求

使用示意图描述不同用户角色的需求。分析确定每个角色的权限及可操作的功能，比如会员可以查看特别信息、修改个人信息、退出登录等；前台管理员能够登录的管理系统，能够发布编辑修改信息，能够审查会员资格等；网站管理员还可以更改栏目、修改网站界面等。制作流程图和示意图将用户需求表现出来；可以让用户参与到示意图的设计中，这样可以正确的反映用户的需求。

3）使用文字和功能体系图描述网站要为用户角色提供哪些服务

根据用户需求，使用文字和功能体系图描述网站要为用户角色提供哪些功能服务

内容。

2. 需求分析报告部分内容实例

1）网站名称

网站名称为“体育用品商店”。

2）网站用户细分

- 个人顾客：体育用品主要面向儿童、青年人和休闲一族。儿童喜欢活泼可爱、年轻人倾向时尚与活力，而中年的休闲族注重品位与身份。
- 团体顾客：随着人们生活水平的提高，在健身娱乐方面表现出强大的需求。在体育用品市场有巨大的消费潜力。在小区健身，全民健身的氛围下，大的团体采购活动已经出现并有进一步升温的趋势。为满足这一市场需要，要求网站为这些大的团体提供便捷的服务。
- 网站新闻管理员：负责网站新闻发布编辑修改工作。
- 网站市场调查管理员：负责市场调查的标题内容发布、编辑、修改等工作。
- 网站维护管理员：负责数据库的管理、页面更新等工作。
- 网站销售管理员：负责网上订单查询、打印订单等工作。

3）用户需求特点分析

用户需求特点可使用示意图来描述，如图 9.1 所示描述了三种顾客的需求特点。

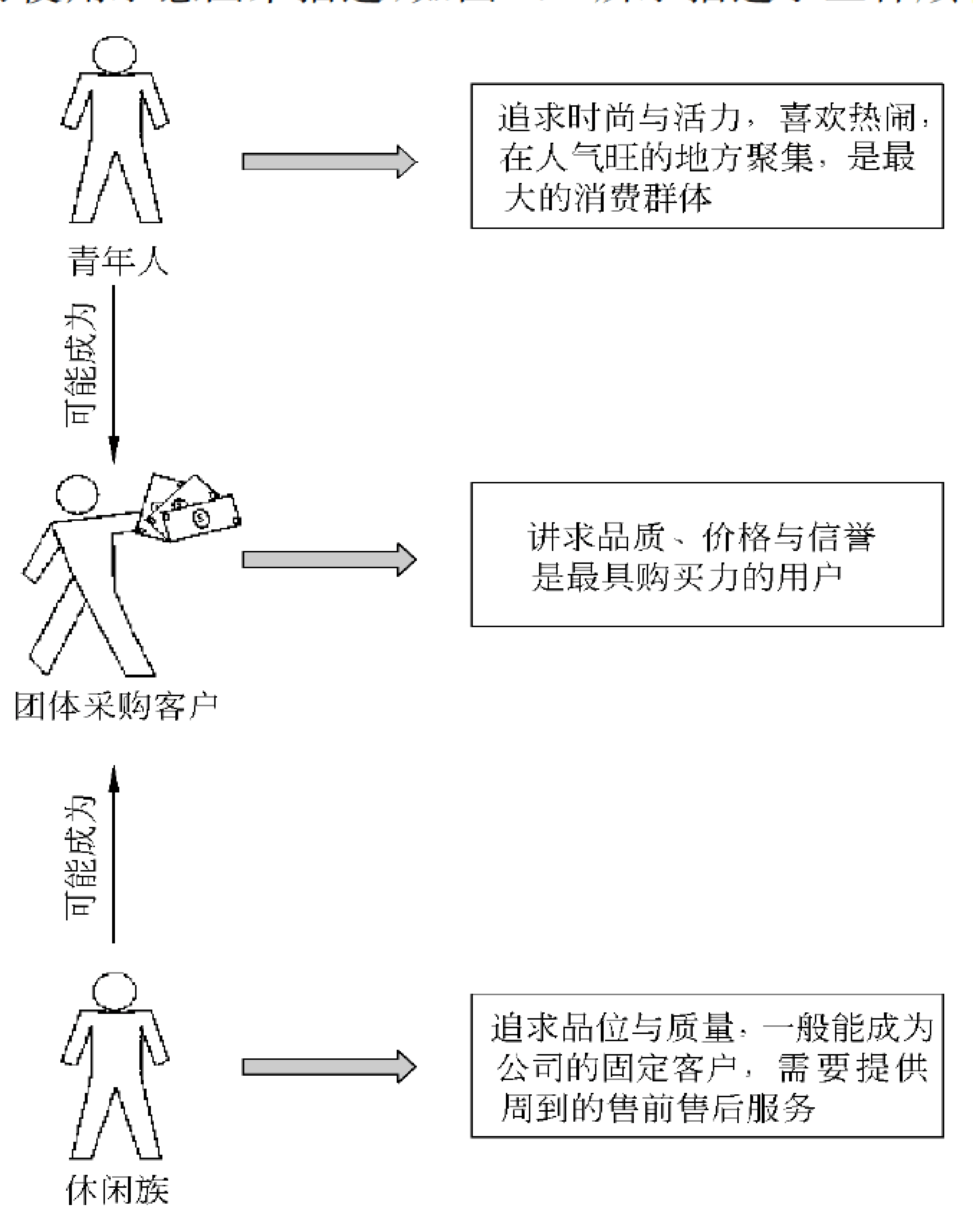


图 9.1　用户需求特点示意图

4）网站为用户提供的服务

可使用文字和功能体系图描述网站要为用户角色提供的服务，如图 9.2 所示描述了提供给顾客、会员的不同服务。

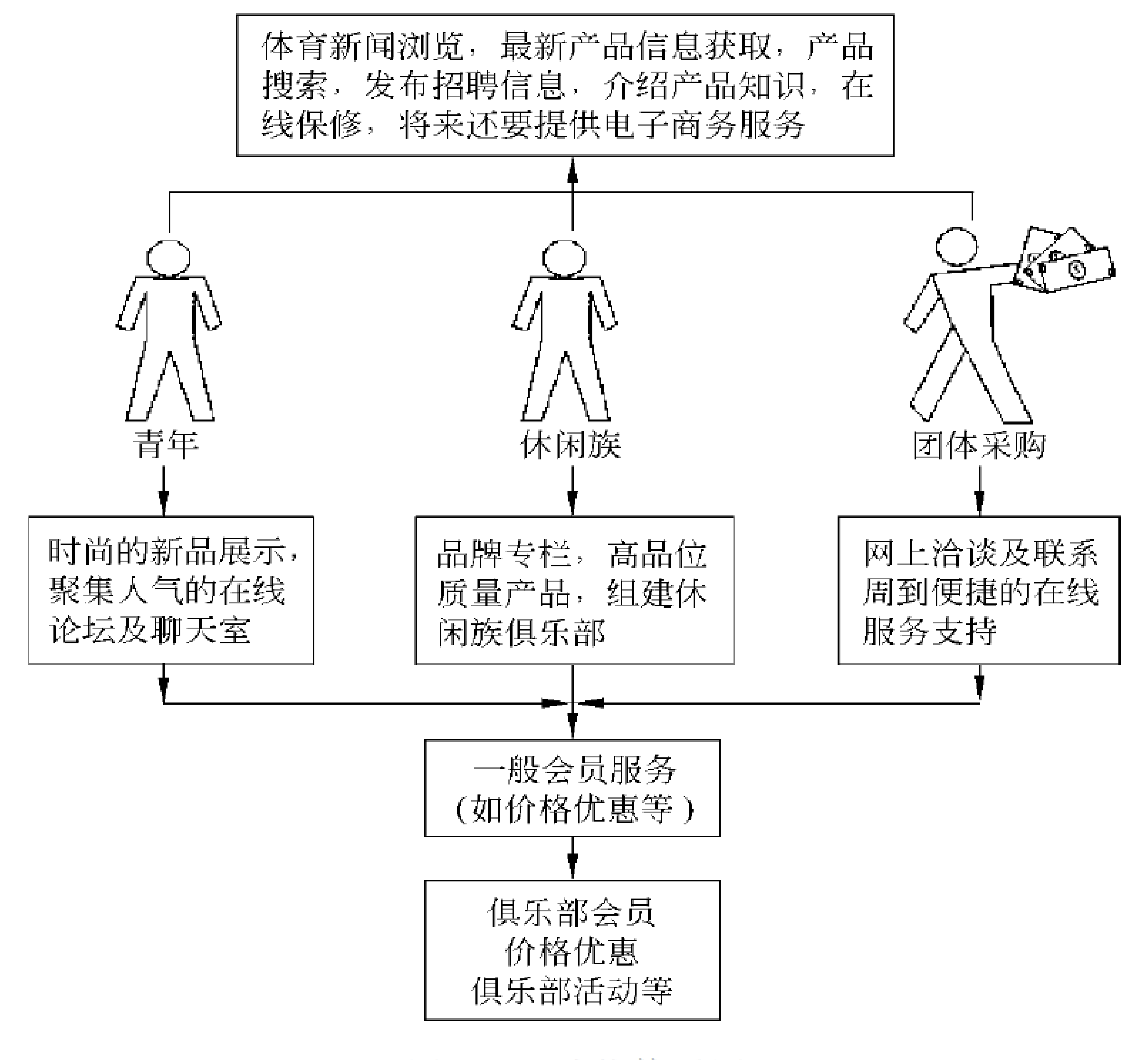


图 9.2　功能体系图

9.4　网站规划阶段

本节的内容主要介绍网站开发规划书的主要内容，它是网站规划阶段的标志性成果。

开发一个网站，离不开事先的详尽规划和精心设计。俗话说得好，“良好的开端是成功的一半”。对于网站开发，所谓的“良好的开端”就是要有一个详尽细致的网站开发方案，这个开发方案就是网站开发规划书，它是所开发网站的一个总体框架。根据这个框架，就可以按部就班地实现整个网站的开发工作，既可以提高效率，又可以节约时间，还有利于规范网站的总体布局，实现统一的网站风格，为创建网站打下良好的基础。

网站规划阶段的工作是在网站开发前面两个阶段的基础上做的工作总结，根据需求分析报告对网站开发建设中所需要的内容、功能、技术、费用、测试、维护等做出书面的、全局的规划。

网站开发规划书主要内容包括：

1. 市场分析

（1）相关网站的市场是怎样的，市场有什么样的特点，是否能够在互联网上开展公司业务。

(2) 市场主要竞争者分析,竞争对手上网情况及其网站规划、功能作用。

(3) 公司自身条件分析、公司概况、市场优势,可以利用网站提升哪些竞争力,建设网站的能力(费用、技术、人力等)。

2. 开发网站的目的及功能类型定位

(1) 确定网站的目的。为什么要建立网站,是为了宣传产品,进行电子商务,还是建立行业性网站?是企业的需要还是市场开拓的延伸?

(2) 确定网站的功能类型。根据公司的需要和计划,确定网站功能类型:产品宣传型、网上营销型、客户服务型、电子商务型等。

(3) 根据网站功能类型,确定网站应达到的目标要求。

3. 网站技术解决方案

根据网站的功能确定网站技术解决方案。

(1) 采用自建服务器,还是租用虚拟主机。

(2) 选择操作系统,用UNIX、Linux还是Window 2000/NT。分析投入成本、功能、开发、稳定性和安全性等。

(3) 采用系统性的解决方案(如IBM,HP等公司提供的企业上网方案、电子商务解决方案)?还是自己开发。

(4) 网站安全性措施,防黑、防病毒方案。

(5) 相关功能程序开发使用什么开发技术,是使用JSP、ASP还是CGI等。

4. 网站内容与功能要求

(1) 确定网站名称

在建站的初期,应给网站起一个好名字。站点名称起着很重要的作用,它在很大程度上决定了整个网站的定位。一个好的名字必须有概括性、简短、要合理、有特色、易记(6个字内),还要符合自己站点的主题风格。例如:新浪网、搜狐、网易。

(2) 根据网站的目的和需求分析划分网站的主要页面内容及包括的主要功能模块

一般企业网站应包括公司简介、产品介绍、服务内容、价格信息、联系方式、网上订单服务等主要页面。

电子商务类网站要提供会员注册、详细的商品服务信息、信息搜索查询、订单确认、付款、个人信息保密措施、相关帮助等功能模块。

(3) 确定网站各功能模块内容概要,可考虑采用网站编程专人负责相关栏目模块内容的准备、开发工作。

(4) 确定网站的层次数,画出网站的结构图。

5. 网站设计要求

(1) 网页美术设计要求。

网站的功能不仅仅局限于传递信息,它在宣传方面的功能丝毫不亚于报刊杂志等传

统媒体，一个优秀的网站和一个实际存在的公司一样，需要整体的形象包装和设计。网页美术设计一般要与企业整体形象一致，要符合 CI 规范，CI(Corporate Identity)就是通过视觉来统一企业的形象。要注意网页色彩、图片的应用及版面规划，保持网页的整体一致性。

(2) 在新技术的采用上要考虑主要目标访问群体的分布地域、年龄阶层、网络速度、阅读习惯等。

(3) 制定网页改版计划，如半年到一年时间进行较大规模改版等。

(4) 确定网站整体风格

开发网站时要根据设计原则和针对访问者确定网站适宜的整体风格。如果浏览者主要是年轻人，页面就要设计得基调明快，活泼富有朝气；如果用户主要群体是政府工作者或科学工作者，那么页面就要设计得严谨、严肃、理性、科学；如果用户主要群体是恋人或从事艺术工作的人，页面就要设计得浪漫并有艺术感；如果用户主要群体是妇女儿童，页面则要设计得温馨、友爱、轻松。

确定网站整体风格后，还要针对不同栏目设计其特有的风格，具体体现在内容安排、链接结构、CI 要素等诸多元素上，要围绕栏目中心思想，用独特的视觉语言进行艺术化的特色设计。

6. 网站维护说明

(1) 服务器及相关软硬件的维护，对可能出现的问题进行评估，制定响应时间。

(2) 数据库维护，有效地利用数据是网站维护的重要内容，因此数据库的维护要受到重视。

(3) 制定内容的更新、调整的时间计划。

(4) 制定相关网站维护的规定，将网站维护制度化、规范化。

7. 网站测试说明

网站发布前要进行细致周密的测试，以保证正常浏览和使用。主要测试内容：

(1) 服务器稳定性、安全性。

(2) 程序及数据库测试。

(3) 网页兼容性测试，如浏览器、显示器。

(4) 根据需要的其他测试。

8. 网站发布与推广计划

(1) 网站测试后进行发布的公关，广告活动计划。

(2) 搜索引擎登记等。

9. 网站建设日程表

10. 网站开发费用预算明细

网站规划书对网站开发建设起到计划和指导的作用，对网站的内容和维护起到定位

作用,也可以作为用户和开发人员之间的协议或合同,是以后网站验收的标准。一个网站的成功与否与建站前的网站规划有着极为重要的关系。

9.5 网站设计阶段

本节主要介绍网站设计报告的具体内容。

9.5.1 网站设计方法

在网站需求分析和网站规划的基础上,接下来可以进行网站设计的具体工作了。

网站设计阶段的主要工作是网站内容和功能的具体设计,要解决如何展示网站浏览的内容、如何实现网站的功能。具体任务包括网站主页结构设计、主要栏目主页设计、确定各页面的内容和功能、网站导航与链接模式设计、网站目录结构设计等。它是进行网站实际制作前的准备工作,网站设计阶段的成果是——网站开发设计说明书,它就像建筑工程的施工图纸,网站设计说明书为开发人员对网站的具体制作提出了具体的接近实际的要求。编写网站设计说明书是网站开发工作中的一个关键环节。

进行网站设计首先要决定采用什么样的方法来进行网站开发设计。和一般软件系统设计的方法类似,网站开发设计也有自顶向下、自底向上和从中间开始等设计方法。

这里介绍网站设计"自顶向下、逐步细化"的结构化、模块化的方法。所谓自顶向下,就是从整个网页的主页(顶),常说的 HomePage 开始设计,向下一层一层地展开。采用这种设计方法,在开始实现网页时可以先用一些空的网页构筑起整个网站的框架,按模块开发相关的应用程序,然后再一步一步地向框架中增加内容,逐步细化,直至最后完成整个网站的制作。

采用这种设计方法的优点是能在总体上统一整个网站的界面风格,使网页的组织结构比较合理。用这种方法,通常一开始就做出一个所谓的"模板",作为以后页面设计的基础,这样就能保证整个网站系统界面的版面风格和功能设置的一致性。

在网站制作时,先将网站的各个栏目作为一个链接放在主页上,以形成网站一个总的概貌,然后分别实现各个栏目的主页和二级页面,以及各个页面中需要的功能模块,最后充实主页上的链接,直到最后实现所有的功能,网站就开发成功了。

网站设计的成果是——网站设计说明书。下面分别介绍网站设计的内容。

9.5.2 网站标题设计

网站标题包括网站名称和 logo 标志。每一个网站都要有一个响亮的名字和独具风格的 logo 标志。例如著名的雅虎、搜狐、新浪等。好的网站名称能简单明了地体现出这个网站的主题,同时具有很好的吸引力。标志也同样要设计得简单而独具特色。网站的 logo 如同商标一样,是网站特色和内涵的集中体现,要让大家看见 logo 就联想起该网站。标志可以是中文,英文字母,可以是符号,图案,可以是动物或者人物等。如新浪用字母 sina+眼睛作为标志,如图 9.3 所示。



图 9.3 新浪网站的标志

9.5.3 网站的内容与功能设计

网站的内容与功能都体现在各个网页上，所以下面主要说明几个主要页面的设计。

1. 首页

首页具有简单、鲜明的特点，吸引浏览者的目光，指引用户到主页。

例如，“网上体育用品商店”网站的首页设计为：网站标题包括一个美术化的网站文字名称和专门设计的图片形式的 logo 标志；一个给浏览者以强烈视觉冲击的 flash 品牌宣传图片；一个进入主页的连接按钮，如图 9.4 所示。

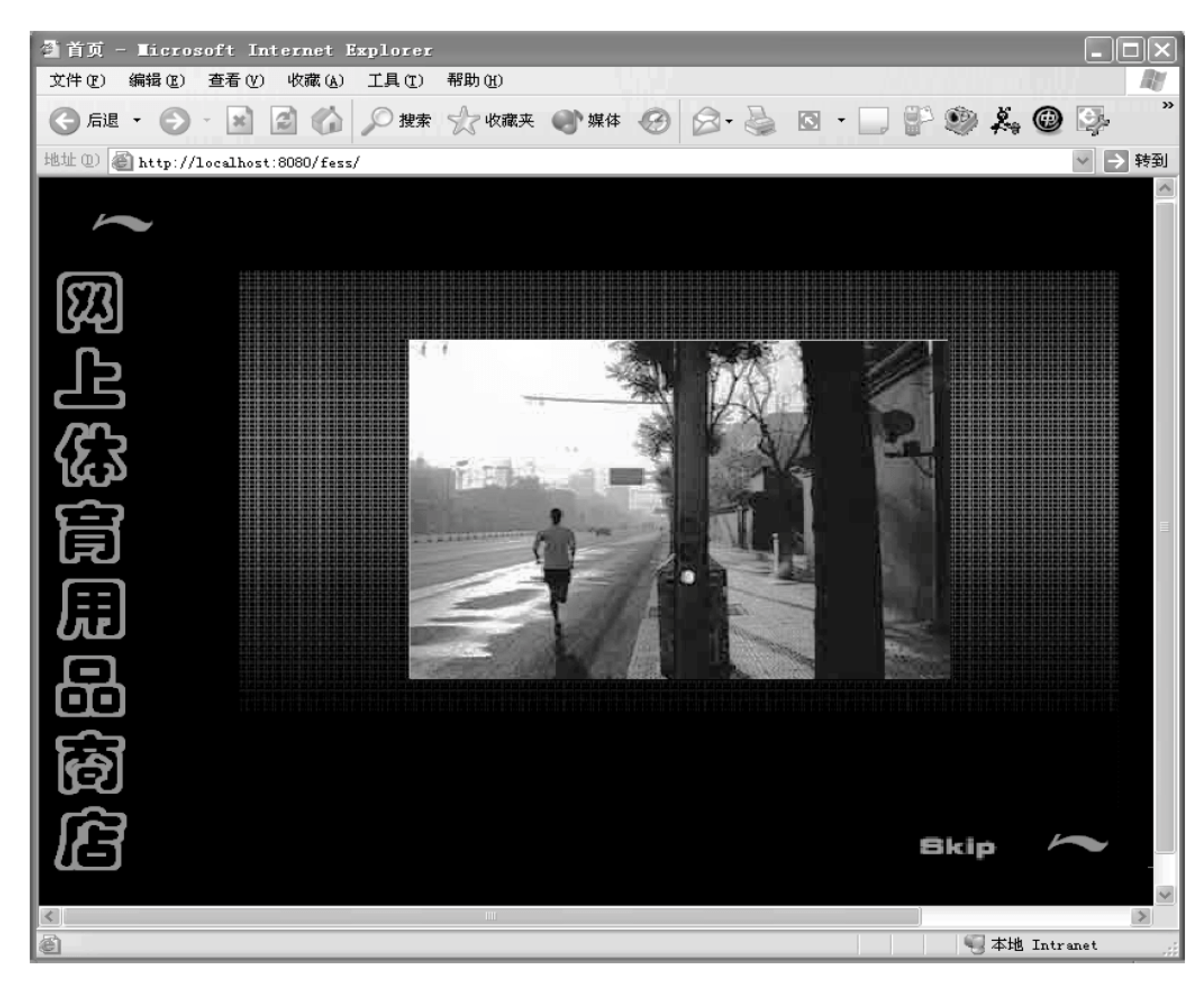


图 9.4 “网上体育用品商店”网站的首页

2. 主页

网站的主页要尽可能地将本站的所有内容和功能栏目展现出来，要最大可能地将主页做成整个网站的缩略图，使得用户只要浏览过主页，就能对网站有一个比较清晰的印象，主页还要提供通向各个主要栏目模块的链接和会员、管理员登录入口。

在主页的风格设计中，色彩的使用也非常的重要。在选择色彩的时候，既不能太过鲜艳，让人觉得刺目难受，也不能过于沉闷，看得让人提不起精神来。

在主页的结构设计中，可将主页划分成几个部分分别设计。这里介绍的是先分为上

中下，再把中间部分分为两块。

（1）主页的上部是网站标题和网站标语图片或主要广告，网站的多数栏目模块的连接也会放在上部，使用户可以很方便在网页上部就能进入他（她）所需要的页面中。

（2）网页的中间部分一般就是网站的各个功能页面的简化版本的组合，如介绍几条最新最有吸引力的新闻，放上几个最新产品介绍，或搞一个简单的分类浏览等等。一般中间部分占页面的比例较大。还可以分为左右两部分。左边为小标题。右边为小标题连接的内容。

（3）网页的下部给出联系方式、版权信息、与其他相关网站的链接和其他内容。

例如，“网上体育用品商店”网站的主页设计：上部——网站标题、列出通向网站各个栏目模块的导航条，为静态内容；中部左上方——提供一个进入功能模块“会员注册和登录”的入口和进入功能模块“管理员维护”的入口，左下方——为由数据库提供的滚动的体育新闻（动态内容）。中部右上方——是一个动态的图片，旨在营造一个动感时尚的氛围，中部右下方——是一个显示欢迎用户的计数器，为动态内容。主页下部——给出联系方式、版权信息和其他链接内容，为静态内容，如图 9.5 所示。



图 9.5 “网上体育用品商店”网站的主页

3. 网站主要栏目设计

主要栏目设计包括各个栏目的主页和二级功能页面的设计。

栏目主页要包括该栏目的主要内容介绍、到各个二级功能页面的接口(链接)。要使浏览者一进入该页面,就能知道本栏目提供哪些服务,可以浏览哪些内容。

二级功能页面是根据内容和功能需要使用的页面,可在制作时具体确定,只要注意提供返回栏目主页面和网站主页的链接即可。

下面以“网上体育用品商店”为例说明一些主要栏目页面的设计。

1) 关于公司栏目

关于公司栏目主页主要包括公司简介、公司文化、公司新闻、诚聘人才的简要内容,并提供到这几个内容的二级页面链接。

公司简介页面介绍公司的概况和发展历程,为静态页面。

公司文化页面宣传公司经营理念,倡导健康时尚充满活力的企业文化,为静态页面。

诚聘人才页面向社会发布最新招聘信息,诚邀有识人才加盟发展,为动态页面。这里要提供一个登记人才信息到数据库的功能。

2) 体育动态栏目

体育动态栏目分为国内国外新闻等页面,还有很多各类赛事的专栏页面,如足球、NBA、体操等。这是本站吸引人气的主要栏目。因而要力求迅速,更新速度要快,各个页面要注明更新日期。让用户浏览最新的国内外体育赛事、发表评论。这里页面比较多,而且经常变换,网站要保持几个栏目的高质量,转载发表高水平的文章,图文并茂,从内容和形式上吸引读者。这里将使用数据库建立一个体育新闻分布、更新系统。

3) 市场信息栏目

第一时间发布最新的市场信息,为动态页面,用户可在此页面向公司提出个人购买和团体购买需求,这里将使用数据库建立一个提交产品订单的系统。

4) 关于产品栏目

关于产品栏目的页面可直接显示新产品发布、服饰类、鞋类、器具类等多个页面,为动态页面。用户可以从中获取各个商品的详细介绍,每个商品都配有图片,图文并茂,这里将使用数据库建立一个提供产品查询、用户“购物车”的系统,方便用户查询和购买产品。

5) 服务中心栏目

该栏目下包括以下几个二级功能页面:

服务热线页面,详细介绍各个分店地址、联系电话等信息,方便用户就近购买,为静态页面。

在线报修页面,包括一个用户可以登记报修信息的数据库系统,公司根据用户提供的信息可以按用户要求的时间提供上门服务。

运动知识页面,向用户提供一些运动过程中的常识和注意事项,为静态页面。

顾客登记页面,包括一个用户可以登记注册成为会员的数据库系统。

产品咨询页面,包括一个产品咨询数据库系统,用户可以查询产品题库中的问题答案。若没有解答,不久后该问题的解答将会发到用户邮箱中。

6) 好友论坛栏目

进入网站为喜爱运动的网友建设的好友论坛,用户可以在里面发表自己的见解,交流心得和体会。包括 BBS、聊天室。

7）会员天地栏目

提供使用会员留言簿、聊天室、参加各种调查活动的入口。

网站主要栏目示意图描述，如图 9.6 所示。

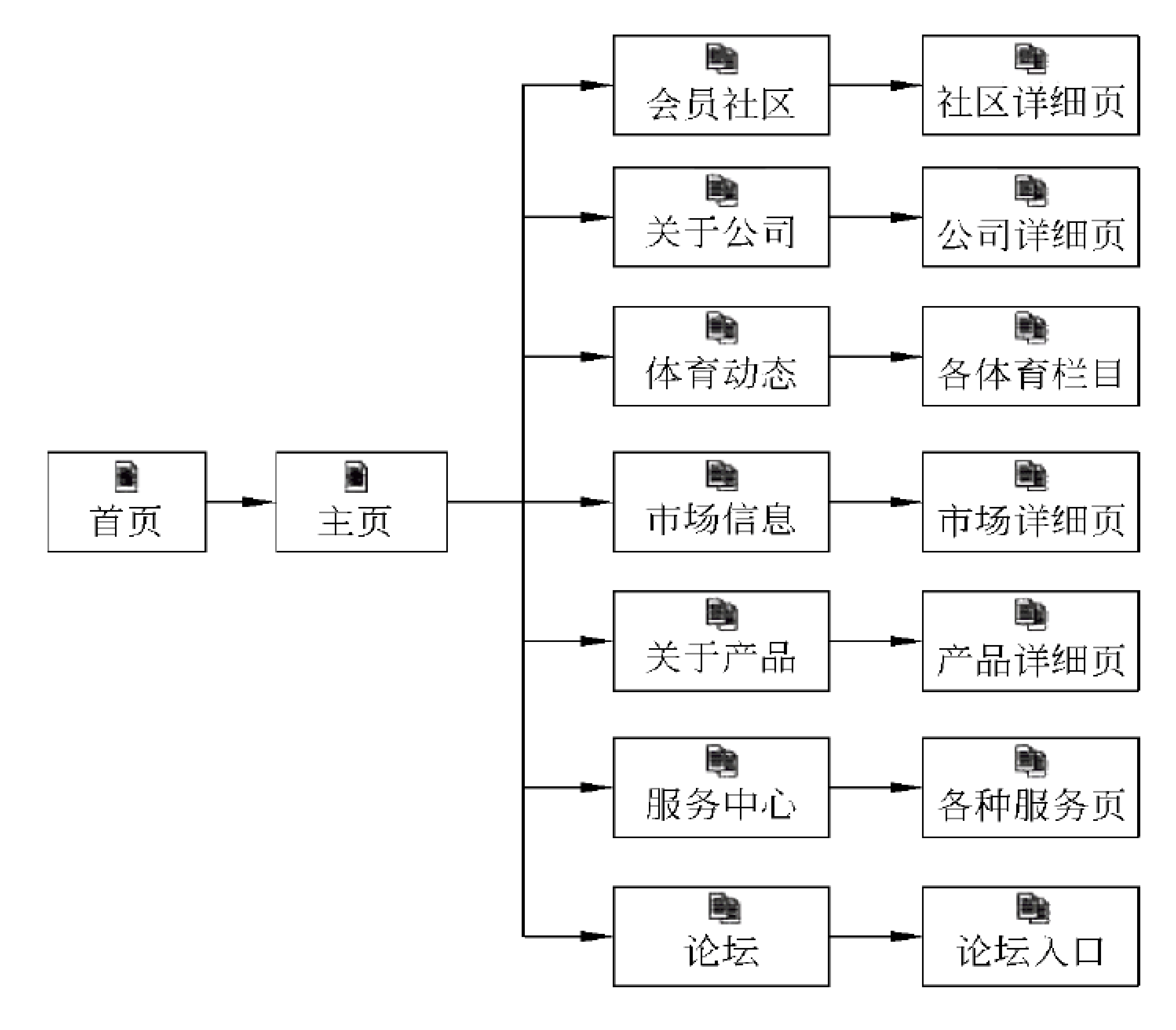


图 9.6　网站主要栏目示意图

9.5.4　目录结构设计

目录结构设计也称为网站物理结构设计，物理结构是指网站文件的物理存储结构，即网站文件在服务器上存储的方式。

由于设计一个网站会涉及许多文件和目录，同时在设计的过程中也会生成许多文件，包括页面文件、程序代码等。如何合理地放置这些文件和目录对于顺利地进行开发工作也是很重要的。特别是在网站的开发初期就合理地规划好目录结构，对于设计网页是很有好处的。如果网站规模很小，网页文件也不多，最简单的方法就是给整个网站建立一个目录，将所有的文件和图片文件都放在该目录下面。复杂的网站可以按网站的内容结构分成不同的部分，每个部分从结构上来说是相对独立的。

例如，“网上体育用品商店”网站的所有网页文件都放在应用服务器下的 fess 目录中，fess 目录下为每个栏目建立一个子目录，以便于未来管理页面的方便。图片统一放在 image 目录中，文件名都使用简短的英文或中文拼音字母命名。

目录结构设计要使用文档资料进行记录，方便查询、增加、维护与使用。

9.5.5　导航与交互设计

1. 导航设计

导航设计要解决的问题是页面位置、去向、路径、返回方法。导航设计有以下类型。

(1) 超文本链接：非顺序的通过超文本链接起来，使用户按使用的顺序和意愿浏览信息。

(2) 导航栏：放置在固定位置的多个标题的超链接。可以是文字或图片。一般有大型导航条、小型导航条、导航目录。

(3) 网站地图：以图形的方式显示站点中所有的页面和栏目部分、内容分类列表等。显示的是网站的逻辑结构，逻辑结构是网站在运行时抽象出来的拓扑结构，它是建立在物理结构之上的。网站地图一般专门提供一个链接的页面。

2. 交互设计

交互设计要解决的问题：解决互相作用、互相交流的方式问题；解决信息传递和选择性问题。人机交互设计要注意交互的简易性(让计算机多做工作)，友好性(主观操作简单交互容易)。还要注意交互的灵活性、明确性、一致性、容错性、反馈性、图形化。

人机交互的常用方式有：问答式、菜单式、功能键、图符、查询语言界面、自然语言界面。要注意菜单层数不能太多，即菜单交互功能深度。菜单宽度不要过大，即同层中菜单项总数。

3. "网上体育用品商店"网站中导航设计

每个页面都包含网站标志，单击网站标志可通向网站主页面，每个页面包含到各个栏目版块的导航条和网站地图，方便用户在本站内浏览，进入各个功能页面。对于超文本链接，用户访问过和未访问的分别用不同的颜色显示。在每个页面设置"返回主页"的链接。

9.5.6 网页版面布局设计

网页版面布局设计统一设计了网站所有页面的布局，而且在设计定型之后一般就很少修改。有时候将网页版面布局设计称为"模板设计"，以后各页面的详细设计和实现都是基于这个模板展开的。

网页版面设计是在限定的面积范围内，合理安排布置图片、图像和文字的位置。版面布局样式主要有"T"字型、"口"字型、"三"字型、"川"字型和自由型，如图 9.7 所示。

例如，"网上体育用品商店"网页版面布局特点是，网页的版面分成三个部分：顶部网站标题与导航条区、中部页面主要内容显示区和底部信息提示区。

除了中部的工作区各个页面不相同外，顶部和底部的内容所有页面基本相同。具体来说，页面的顶部包含了网站的标题(各个主要栏目页面可显示不同标题名称)；网站导航栏(所有页面都相同)，通过导航栏可以进入到网站的任何一个部分；底部区包括的内容比较简单，如返回首页链接和一些版权信息等。而对于页面的中部显示区不同的页面差别就很大了。如网站首页显示的内容比较全面，它几乎包含了本网站所有内容的缩小版本。其他功能页面可能显示的是查询的结果、产品介绍等。

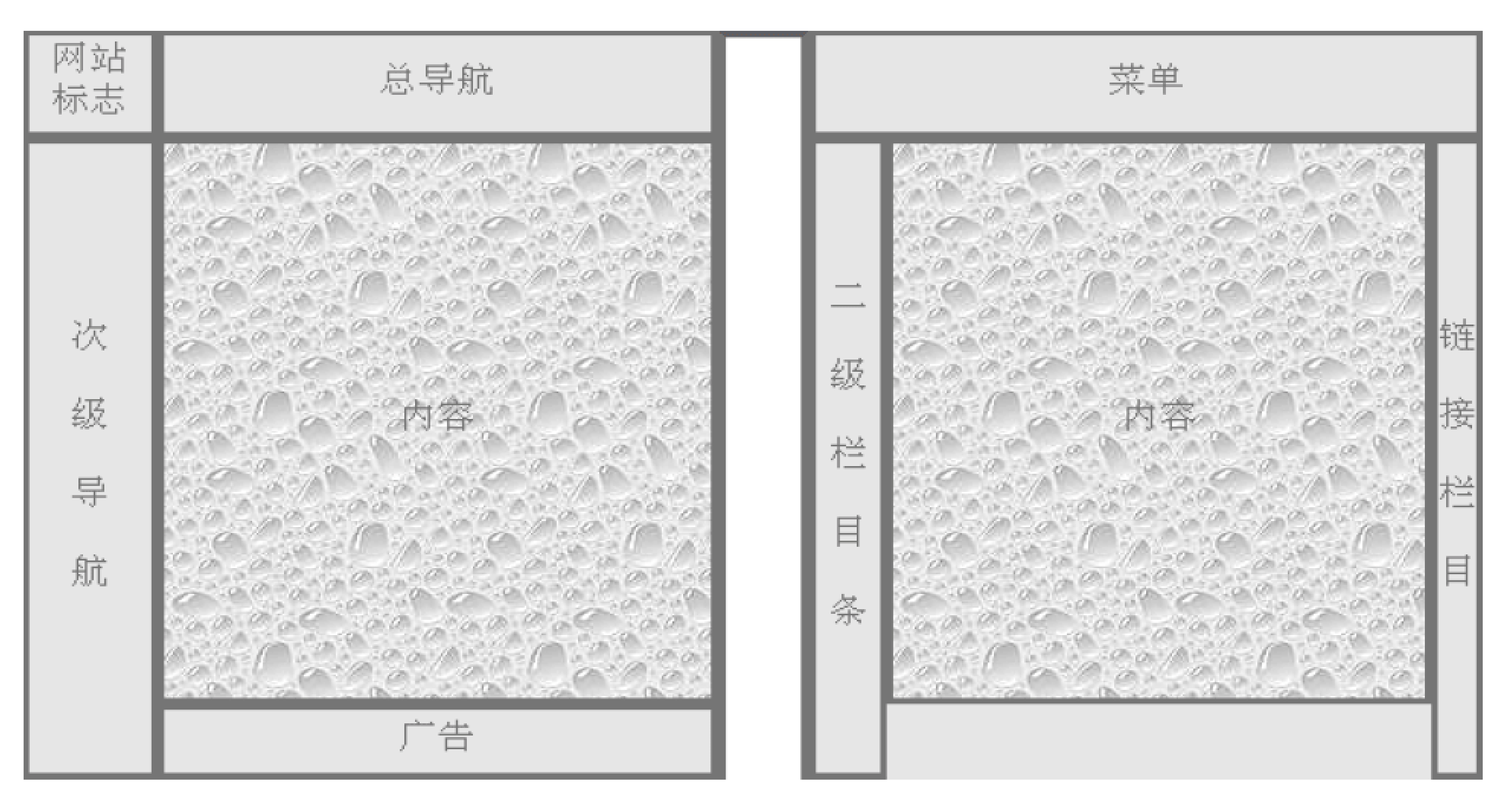


图 9.7 版面布局样式

9.5.7 网站外观设计

网站在宣传方面的功能丝毫不亚于报刊杂志等传统媒体，一个优秀的网站和一个实际存在的公司一样，需要整体的形象包装和设计。网站外观设计包括网页色彩设计、CI设计、版面设计、风格与创意设计等。如使用统一的项目符号，统一的网站标志等。

1. 网站色彩设计

网站色彩设计非常重要，打开一个网站，给用户留下第一印象的既不是网站丰富的内容和完善的功能，也不是网站合理的版面布局，而是网站的色彩。色彩对人的视觉效果非常明显，一个网站设计成功与否，在某种程度上取决于设计者对色彩的运用和搭配。

因为网页设计属于一种平面效果设计，在排除立体图形、动画效果之外，在平面图上，色彩的冲击力是最强的，它很容易给用户留下深刻的印象。因此，在设计网页时，必须要高度重视色彩的搭配。

要合理地选取页面的背景色，而且一旦选定了网站的主背景色，所有其他色彩的使用都要围绕它展开，要选择那些能很好地融入到主背景色中的颜色，决不要使用与之相对立的颜色。

一个网站不可能单一的运用一种颜色，让人感觉单调，乏味；但是也不可能将所有的颜色都运用到网站中，让人感觉轻浮，花哨。一个网站必须有一种或两种主题色，不至于让客户迷失方向，也不至于单调，乏味。一个页面尽量不要超过 4 种色彩，用太多的色彩让人没有方向，没有侧重。当主题色确定好以后，考虑其他配色时，一定要考虑其他配色与主题色的关系，要体现什么样的效果。

2. 网站的 CI 设计

网站的 CI 设计包括以下内容：

（1）网站标志。

（2）网站标准色彩，标准色彩是指能体现网站形象和延伸内涵的色彩。一个网站的标准色彩不超过3种，太多则让人眼花缭乱。标准色彩要用于网站的标志，标题，主菜单和主色块。给人以整体统一的感觉。至于其他色彩也可以使用，只是作为点缀和衬托，绝不能喧宾夺主。例如IBM网站的深蓝色，肯德基网站的红色条型，Windows视窗标志上的红蓝黄绿色块，都使人觉得很贴切，很和谐。

（3）网站标准字体，和标准色彩一样，标准字体是指用于标志，标题，主菜单的特有字体。一般网页默认的字体是宋体。为了体现站点的"与众不同"和特有风格，可以根据需要选择一些特别字体。

（4）网站宣传标语，网站的宣传标语可以说是网站的精神，网站的目标。用一句话甚至一个词来高度概括。类似实际生活中的广告金句。例如：鹊巢的"味道好极了"；麦斯威尔的"好东西和朋友一起分享"；Intel的"给你一个奔腾的心"。

以上四方面：标志、色彩、字体、标语是一个网站树立CI形象的关键，确切的说是网站的表面文章，设计并完成这几个任务，网站将脱胎换骨，整体形象会大大提高。色彩与字体的设计可以专门使用style样式文件来保存，以利于制作各个页面时使用。

3. 版面设计

所谓版面设计，就是在版面上将有限的视觉元素进行有机的排列组合，将理性思维个性化地表现出来，是一种具有个人风格和艺术特色的视觉传达方式。它在传达信息的同时，也产生感官上的美感。无论是进行电视广告、报纸广告、杂志、包装、网页等设计，都需要版式设计。

版面设计是现代设计艺术的重要组成部分，是视觉传达的重要手段。表面上看，它是一种关于编排的学问；而实际上，它不仅是一种技能，更是技术与艺术的高度统一。一幅优秀的网页作品，它有了好的创意和内容，也就等于一个人有了好的灵魂。版面设计就好比这个人美妙的身材，二者相互影响与融合，才能使观者视线久久难移，产生震撼力。

网页的版面设计要主次分明，中心突出。在一个网页上，必然要考虑视觉的中心，这个中心一般在页面的中央，或者在中间偏上的部位。因此，一些重要的文字和图片一般可以安排在这个部位，在视觉中心以外的地方就可以安排那些稍微次要的内容，这样在页面上就突出了重点，做到了主次有别。

在网页中，主题要成为视觉中心，吸引人的视线，主题不鲜明突出，会扰乱观者的视觉流程，使网页版面显得散乱，破坏其和谐。一幅主题不鲜明不突出的网页，就好比一堆零乱的垃圾，无法让冲浪者享受到美。版面设计的最终目的是使网页产生清晰的条理性，用悦目的组织来更好地突出主题，达成最佳的表达效果。

网页的版面设计要注意文字与图片的布局。网页作为一种版面，既有文字，又有图片。文字有大有小，还有标题与正文之分；图片也有大小，而且有横竖之别。图片和文字都需要同时展示给观众，不能简单地罗列在一个页面上，这样往往会搞得杂乱无章。必须根据内容的需要，将这些图片和文字按照一定的次序进行合理编排和布局，使它们组成一个有机的整体，展现给广大的观众。文字和图片还具有一种相互补充的视觉关系，一个网

页页面上的文字太多,就显得沉闷,缺乏生气;页面上图片太多,缺少文字,就会减少页面的信息容量。因此,最理想的效果是文字与图片的密切配合,互为衬托,既能活跃页面,又使网页有丰富的内容。

网页的版面设计要注意大小搭配,相互呼应。较长的文章或标题,不要编排在一起,要有一定的距离;同样,较短的文章,也不能编排在一起。对待图片的安排也是这样,要互相错开,造成大小之间有一定的间隔,这样可以使页面错落有致,避免重心的偏离。

4. 网站的整体风格和创意设计

风格(style)是抽象的。风格包括站点的CI(标志、色彩、字体、标语)、版面布局、浏览方式、交互性、文字、语气、内容价值、存在意义、站点荣誉等诸多因素。

风格是独特的,是站点不同于其他网站的地方。或者色彩,或者技术,或者是交互方式,能让浏览者明确分辨出这是你的网站独有的。

风格是有人性的。通过网站的外表、内容、文字、交流可以概括出一个站点的个性、情绪,是温文儒雅,还是执著热情;是活泼易变,还是放任不羁。

风格是建立在有价值内容之上的。例如,希望站点给人的印象是专业的物流电子商务网站,应该有创意,有技术实力,有美感,有冲击力,可以信赖,信息最快,交流方便。

创意(idea)是网站生存的关键。实质上,创意是传达信息的一种特别方式。创意并不是天才者的灵感,而是思考的结果。创意是将现有的要素重新组合。创意的目的是更好的宣传推广网站。

例如,"网上体育用品商店"网站在外观设计中在大多有动态的网页中一致使用黑色的背景,在此背景下通过各种颜色的搭配,表现动感的运动之美。

在版面设计上,运用flash,动态图片等动感视觉元素有机结合,努力向浏览者传达健康、向上、充满活力的运动理念。图像flash的制作要讲求创意富有艺术效果,将运动之美融入到画面中,从而给浏览者以强烈的视觉冲击。

网站的图片和flash设计要紧紧围绕健康时尚的运动主题。做到主次分明,中心突出。由于"网上体育用品商店"网站在建设初期主要注重宣传效果,因而将表现力主要体现在突出企业形象上。首页中心放一个flash的宣传动画片,在主页上将设计一个表现体育产品的动态图片,突出强调网站的运动主题。

在各栏目的网页外观设计上注重图文并茂,相得益彰。形成鲜明活泼的整体风格。

9.5.8 页面详细设计

1. 页面详细设计内容(注意要和需求分析说明相结合)

(1) 该页面的作用;

(2) 该页面要反映的具体内容;

(3) 该页面要使用的图片、动画、声音文件的内容、格式等要求;

(4) 确定静态和动态内容、确定动态功能模块使用什么Web开发技术工具进行开发。

2. 体育用品网站主要页面详细设计实例

1）首页

首页的作用是吸引浏览者，指引用户到主页。反映的具体内容包括黑色背景，网页左边是网站名称和标志。网页中心是一个 flash 品牌宣传片，在 flash 下面有一个小箭头图片，用户单击后可进入网站主页面。首页使用一个 flash 动画文件、一个箭头图片。

该页面没有功能模块。使用 HTML 技术。

2）主页

主页的作用是让用户了解该网站的主要内容、功能和服务。

反映的具体内容与素材：黑色的背景，页面上方网站动态图片 logo 和使用 JavaScript 等技术编写的导航栏。中部左方分为两部分：会员注册登录区和管理员登录入口、滚动的新闻标题；中部右上方是一个动态的 GIF 图片，一些动感的图片和背景协调地搭配在一起。营造出一种动感时尚的氛围；中部右下方是显示欢迎用户的计数器；最下面是联系方式和一些链接信息。

该页面有几个动态功能应用，分别是会员注册、登录和管理员入口、记录访问网站人数的计数器和滚动新闻标题及链接的模块，该页面要使用 JSP 等技术来编写这些动态内容。

3）体育动态栏目主页

该页为体育动态栏目的主页，作用是及时反映国内外体育动态。具体内容布局：中部左边是各个热门版块比如足球、NBA 等的栏目链接。中部右上方采用滚动方式显示最新的国内国外新闻。中部右下方是几个主打专栏的新闻信息，在热门新闻前加醒目标志提示，后面注明发布时间，中间放些图片，让网站形式更丰富一些。页面上漂浮一些图像链接，可单击进入相关网站。

发布新闻讲求时效性，因而后台需要一个新闻发布数据库系统进行支持，这里使用 JSP 和 MySQL 数据库技术开发新闻发布子系统。

9.5.9 数据库设计

网站涉及许多重要数据，这些数据中包括产品信息、销售信息、用户信息等内容，它们对于网站的建设和发展有着非常重要的作用，对这些数据要完成保存、查询和加密的工作，使用数据库是最好的解决办法。

1. 数据库与表的设计

建立网站的时候，建立多少个数据库，每个数据库中有多少张表，开发人员要根据所建网站的实际情况来决定。这之中有数据量大小的考虑，有数据更新频率和数量的考虑，有数据库查询操作的频繁程度的考虑，还有管理层次上的考虑，等等，可以说需要注意和分析的问题是非常多的。实际操作之前必须经过慎重的设计工作。

数据库系统中的数据是企业了解用户的主要渠道，是企业的宝贵资源，一定要充分利用和重视。数据库系统不仅要有存取信息的功能，还要注意数据库的安全性和条理性。

最大限度获取和使用用户信息。

2. 数据结构规范化

数据库设计的关键是设计数据表中的数据项(字段),要符合数据结构的规范化要求,最大限度地避免数据冗余。

一般的数据结构要满足以下要求:

(1) 在表(二维表)中任意一列上,数据项应属于同一属性,即不允许有重复数据项。

(2) 在表中所有行都是不相同的,不允许有重复的行。

(3) 在表中,行的顺序无关紧要。

(4) 在表中,列的顺序无关紧要,但不能有重复的列。

(5) 每个数据表必须有一个或多个数据项组合构成的主关键字,其他数据项元素要由主关键字来确定。

(6) 非主关键字的数据项元素之间相互独立,没有函数传递依赖关系。

9.6 网站开发硬件和软件准备

本节主要介绍在网站具体实现前的软硬件准备工作。

9.6.1 硬件准备

要创建一个网站,需要有一台安装了 Web 服务器的计算机。但不是任何计算机都能担当网站服务器的重任,这是因为对于计算机的要求还是比较高的。这个要求还和具体的网站规模、网站的访问量有关。一般来说,最小型的个人网站规模大概是在任一时刻的访问量不超过 5 个,这样的网站建立在 Pentium 4、128MB 内存的计算机上就能很好地工作。所使用的服务器软件也是比较小型的。如果是在一个主机超过四五十台的局域网上建立网站,所需的服务器至少得有 128MB RAM,CPU 工作频率应在 300MHz 以上。

可以把网站建在比较高级的工作站上,如管理本地局域网的服务器,甚至可以把网页上传到提供个人主页空间服务的全国性的大型主机上,这样就能保证网页有很好的硬件环境。也可以在自己的机器上建立一个小型的站点,用于本地自己主页的测试。

9.6.2 软件准备

首先,要有 Web 服务器软件。现在流行的基于 PC 的 Web 服务器有如下几个:

(1) Windows 提供的 Personal Web Server 和 IIS。

(2) Linux 的 Apache。

(3) 支持 JSP 的 Tomcat。

为编制网页代码,可准备一个能自动编写静态内容代码的网页制作工具,如 FrontPage 等。因为动态内容主要采用手工编写代码方式,可使用 EditPlus 编辑器或 Eclipse。

9.7　网站开发素材的收集和制作

在具体开发网站前，要先进行素材的准备和收集。要收集各个网页需要的文字素材和图片素材。还要根据需要由开发人员自己制作一些特殊的图片、动画文件、flash 文件、菜单图片、文字图片等。这部分工作也需要花费大量的时间，网站的内容和外观和这部分工作有直接的关系。前面所学的网站编程技术，只是提供一个通过浏览器和服务器中显示这些内容的方法。这部分工作可以由专人负责进行收集和制作。

在网站制作实现阶段可以在项目开发组中按不同技术或不同栏目与功能模块分工完成网站网页的制作任务、测试任务。

当网页制作与测试任务完成后，要将开发的网站发布到某个 Web 服务器中，使用户浏览与使用，此时网站开发进入到网站宣传推广阶段，要通过各种方式与方法向亲朋好友推荐，在搜索引擎登记，在新闻组和 BBS 上作宣传等。

网站发布后即进入到网站运行维护阶段，需要对网页进行日常维护，包括不断测试网络，宣传与推广网站，对网站进行优化与更新，对网站进行备份与恢复。

9.8　思考与练习

9.8.1　思考题

9-1　为什么要把网站开发过程分为多个阶段？

9-2　为什么要编写多种网站开发报告？

9.8.2　实践练习

9-1　成立开发网站项目小组：确定项目组长、项目组成员分工、讨论确定网站开发项目。

9-2　编写调查表，对不同的网站用户进行需求调查，并编写调查报告。

9-3　调查不同类型的网站，写出调查报告，包括网站名称、网站地址、网站特色，分析它们在页面内容、功能、结构、导航与交互、布局、色彩、CI、整体风格等方面的优缺点。

9-4　编写网站需求分析报告。

主要内容包括：

(1) 初步确定的网站名称；

(2) 确定网站的用户角色(即哪些人员浏览和使用网站)；

(3) 制作流程图或示意图描述不同用户角色的所有需求；

(4) 使用文字和功能体系图描述网站要为用户角色提供什么服务。

9-5　编写网站开发规划书，网站开发规划书主要内容：

(1) 网站开发背景；

(2) 建设网站目的及功能定位；

（3）网站技术解决方案；
（4）网站内容与功能规划；
（5）网页设计要求；
（6）网站测试说明；
（7）网站发布说明；
（8）网站维护说明。
9-6　组成网站开发小组，编写网站设计说明书。内容包括：
（1）网站标题设计；
（2）网站内容与功能设计（按页面和功能模块）；
（3）网站目录结构设计；
（4）导航与交互设计；
（5）网页版面布局设计；
（6）网站外观设计；
（7）主要页面详细设计（首页、一级页面、二级页面）；
（8）数据库设计。



第10章 网站开发应用实例

本章将引导读者进入网站开发的制作阶段，通过“网上体育用品商店”网站的开发，掌握制作导航条、计数器模块、会员登录和注册应用程序模块、滚动新闻程序模块、模块化的主页、用户留言应用程序的方法，完成一个网站首页、主页、主要二级页面、功能模块相关页面的开发任务。

本章主要解决以下问题：

- 如何使用公用文件；
- 制作首页有哪些注意事项；
- 制作一个模块化的主页有哪些注意事项；
- 如何制作动态功能模块。

10.1 制作网站共用文件

本节主要介绍网站通常使用的公用文件。

创建“网上体育用品商店”网站时，可先编写网站共用的程序文件，例如样式表文件、连接数据库的文件、公用的函数文件、公用的导航栏、公用的顶部、底部页面文件等，使用这些公用的文件可以统一网站的风格，提高代码重用性。

10.1.1 共用的 CSS 样式表文件

1. 程序文件功能说明

根据网站设计说明书的要求创建 style.css 样式表文件，其功能是统一确定网站页面的背景、主题色彩、文字属性、链接、位置、颜色等内容。

2. style.css 程序代码

下面给出 style.css 样式表文件的源代码，可参考使用。

```
<!--链接的颜色和特性-->
a:link{color:green;text-decoration:none}
```

```
a:visited{color:red;text-decoration:none}
a:hover {color: #ff00ff;text-decoration:underline}
<!--背景颜色和背景图片-->
body{background-color:"#FFFFFF"; background-image:"image/di.gif";}
<!--网站名称的样式-->
.bt{color:red;font- family:华文彩云;font-size:40pt;}
<!--注意文字的样式-->
.zhuye {
 font-family: "楷体_GB2312";
 font-size: small;
 font-style: normal;
 text-decoration: blink;
 background-image: url('????????/????-????????.files/ln_news.gif');
}
<!--新闻文字的样式-->
.news {
 font-family: "楷体_GB2312";
 color: #FFCC00;
 text-decoration: underline;
}
<!--向用户表示欢迎文字的样式-->
.welcome {
 font-size: 1px;
 color: #99ccff;
 background-color: # 999999;
}
<!--对象在页面的位置样式-->
.Layer0{
position:absolute; width:160px; height:115px; z-index:11; left: 155px; top: 175px;
}
.Layer1{
position:absolute; width:160px; height:215px; z-index:11; left: 10px; top: 400px;}
.Layer2{
position:absolute; width:400px; height:50px; z-index:11; left: 280px; top: 630px;}
.Layer3{
position:absolute; width:366px; height:116px; z-index:12; left: 270px; top: 300px;}
.Layer4{position:absolute; z-index:12; left: 320px; top: 320px;}
.Layer5{position:absolute; z-index:12; left: 10px; top: 660px;}
```

注意：style.css 文件可以在需要时不断添加需要的标记。开始时可以只是包含共同使用的一些基本标记。

10.1.2 统计登录网站人数的计数器程序文件

1. 程序功能说明

有的计数器不够科学，只要刷新页面计数器的值就会加 1。计数器程序的功能是在

用户第一次登录网站时刷新一次计数器的值,但再次刷新页面时不添加计数,除非关闭浏览器窗口再打开时才会添加计数。

在浏览器地址栏输入 http://localhost:8080/myjsp/jsq2jsp,计数器程序运行结果如图 10.1 所示。



图 10.1 计数器程序运行结果

2. jsq2.jsp 程序代码

下面给出 jsq2.jsp 计数器程序文件的源代码,可参考使用。

```
<%
String s= (String)application.getAttribute("count");
String id1=session.getId();
String u="";
     if(s==null)
        //设置 application 的 count 属性值为 1
        { application.setAttribute("count","1");
     out.println("欢迎你!你是本网站第"+1+"位访问者。<br><br>");}
     else
        {
          if(session.isNew())
          {
          s=(Integer.parseInt(s)+1)+"";
          application.setAttribute("count",s);
          out.println("欢迎你!你是本网站第"+s+" 位访问者。<br><br>");
          }
          else{
          out.println("欢迎你!你是本网站第"+s+" 位访问者。<br><br>");
        }
        }
%>
```

注意:在计数器 jsq2.jsp 程序中不仅使用了 jsp 技术中的 application 对象,还使用了 session 对象,使用 application 的 count 属性来记录访问该网站的人数,根据 session.

isNew()来判断用户是不是新登录该网站。

10.1.3 导航栏程序文件

1. 程序功能说明

导航栏程序文件的功能是不仅可以链接到网站各个栏目的主页页面，还可以直接链接到各个栏目的二级页面。导航栏可以在网站的其他页面中使用，主要用在主页中。图 10.2 显示的是该导航栏的显示结果。当鼠标指向栏目标题时会显示该栏目下的主要功能页面，鼠标离开标题，下面的小标题会隐藏起来。

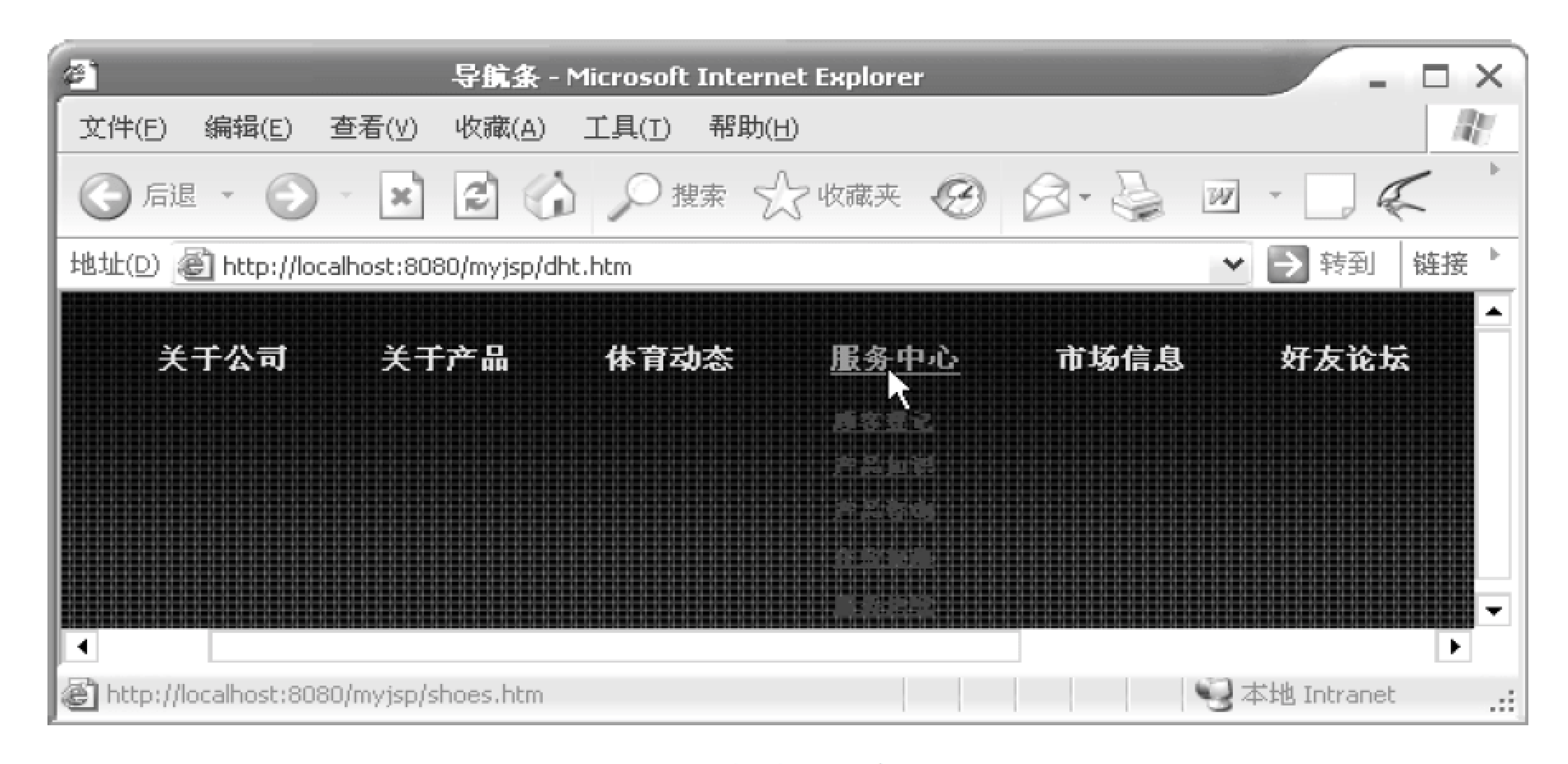


图 10.2 导航栏程序运行结果

2. 创建导航栏使用的功能函数文件 dht.js

导航条具有一个动态功能，即通过滑动鼠标来显示和隐藏某个命名区域内的内容。功能是使用 JavaScript 脚本语言实现的。

为了使程序代码清晰，将创建动态功能的脚本与创建页面格式的 html 代码分开，将 JavaScript 功能函数单独存放在以 js 作为后缀名的文件中。这种方式可以提高程序的重用性，使程序模块化。其他文件需要时可以直接引用 js 文件中的函数。

下面给出 JavaScript 的功能函数文件 dht.js 的源代码，可参考使用。

```
function MM_findObj(n, d)
{
  var p,i,x;
  if(!d)d=document;
  if((p=n.indexOf("?"))>0&&parent.frames.length)
 {
    d=parent.frames[n.substring(p+1)].document;
 n=n.substring(0,p);
 }
  if(!(x=d[n])&&d.all) x=d.all[n];
```

```
  for(i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++)x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById)x=d.getElementById(n);
  return x;
}

function MM_showHideLayers()
{
  var i,p,v,obj,args=MM_showHideLayers.arguments;
  for (i=0;i<(args.length-2); i+=3)if((obj=MM_findObj(args[i]))!=null)
    {v=args[i+2];
    if(obj.style)
        {
        obj=obj.style;
        v=(v=='show')?'visible':(v=='hide')?'hidden':v;
        }
    obj.visibility=v;
    }
}
```

3. dht.htm 程序代码

下面给出导航栏程序文件 dht.htm 的源代码,可参考使用。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>导航条</title>
<script src="dht.js" language="JavaScript" type="text/javascript">
</script>
<link href="css/style.css" rel="stylesheet" type="text/css">
</head>

<body bgcolor="#FFFFFF" background="image/di.gif">
<div id="Layer3" style="position:absolute; width:582; height:39; z-index:4; left: 100;
top: 15;">

  <table width="582" height="36" border="0">
    <tr>
      <td width="94" height="32" onMouseOver="MM_showHideLayers('gs','','show')"
onMouseOut="MM_showHideLayers('gs','','hide')">
        <div align="center">
      <font color="#CCCCCC" size="2">
      <strong>关于公司</strong></font></div></td>
      <td width="94" onMouseOver="MM_showHideLayers('cp','','show')"onMouseOut="MM_
```

```
showHideLayers('cp','','hide')">
          <div align="center"><font color="#CCCCCC" size="2">
        <strong>关于产品</strong></font></div></td><td width="94">
          <div align="center"><font color="#CCCCCC" size="2">
            <strong onMouseOver="MM_showHideLayers('ty','','show')" onMouseOut=
"MM_showHideLayers('ty','','hide')">体育动态
          </strong></font></div></td>
        <td width="94" onMouseOver="MM_showHideLayers('fw','','show')" onMouseOut="MM_
showHideLayers('fw','','hide')">
          <div align="center">
        <font color="#CCCCCC" size="2">
        <strong><a href="shoes.htm">服务中心<a>
        </strong></font></div></td>
        <td width="94" onMouseOver="MM_showHideLayers('sc','','show')" onMouseOut="MM_
showHideLayers('sc','','hide')">
          <div align="center">
        <font color="#CCCCCC" size="2">
            <strong onMouseOver="MM_showHideLayers('sc','','show')" onMouseOut=
"MM_showHideLayers('sc','','hide')">市场信息
          </strong></font></div></td>
        <td width="94" onMouseOver="MM_showHideLayers('bbs','','show')" onMouseOut="MM_
showHideLayers('bbs','','hide')">
          <div align="center">
        <font color="#CCCCCC" size="2">
            <strong onMouseOver="MM_showHideLayers('bbs','','show')" onMouseOut=
"MM_showHideLayers('bbs','','hide')">好友论坛
          </strong></font></div></td>
      </tr>
    </table>
</div>
        <div id="gs" style="position:absolute; width:74; height:51; z-index:9; left: 118;
top: 50; visibility: hidden">

          <table width="72" border="0" onMouseOver="MM_showHideLayers('gs','','show')"
onMouseOut="MM_showHideLayers('gs','','hide')">
            <tr><td class="company">
          <strong><font color="#CCCCCC" size="1">
          <a href="clothe.htm">公司简介</a></font>
          </strong></td></tr>
            <tr><td class="company">
          <strong><font color="#CCCCCC" size="1">
          <a href="shoes.htm">公司文化</a></font></strong></td></tr>
            <tr><td class="company">
          <strong><font color="#CCCCCC" size="1">
```

```
<a href="tool.htm">招聘信息</a></font></strong></td></tr>
</table>
</div>

<div id="ty" style="position:absolute; width:85; height:57; z-index:8; left:
309; top: 48; visibility: hidden">

<table width="85" border="0" onMouseOver="MM_showHideLayers('ty','','show')"
onMouseOut="MM_showHideLayers('ty','','hide')">
<tr><td class="company">
<strong><font color="#CCCCCC" size="1">国内体育
</font></strong></td></tr>
<tr><td class="company">
<strong><font color="#CCCCCC" size="1">国际体育
</font></strong></td></tr>
<tr><td class="company"><strong><font color="#CCCCCC" size="1">奥运 2008
</font></strong></td></tr>
</table>
</div>

<div id="cp" style="position:absolute; width:74; height:51; z-index:9; left: 216;
top: 49; visibility: hidden">

<table width="72" border="0" onMouseOver="MM_showHideLayers('cp','','show')"
onMouseOut="MM_showHideLayers('cp','','hide')">
<tr><td class="company"><strong>
<font color="#CCCCCC" size="1">
<a href="clothe.htm">
服饰类</a></font></strong></td></tr>
<tr><td class="company"><strong>
<font color="#CCCCCC" size="1">
<a href="shoes.htm">鞋类</a>
</font></strong></td></tr>
<tr><td class="company">
<strong><font color="#CCCCCC" size="1">
<a href="tool.htm">
器具类</a></font></strong></td></tr>
</table></div>

<div id="bbs" style="position:absolute; width:88; height:38; z-index:10; left:
612; top: 50; visibility: hidden">

<table width="88" border="0" onMouseOver="MM_showHideLayers('bbs','','show')"
onMouseOut="MM_showHideLayers('bbs','','hide')">
```

```
<tr><td height="14" class="company" width="82">
<font color="#CCCCCC" size="1">
<strong>
BBS</strong></font></td></tr>
<tr><td class="company" width="82">
<font color="#CCCCCC" size="1"><strong>
留言板</strong></font></td></tr>
</table></div>
<div id="sc" style="position:absolute; width:486; height:50; z-index:7; left:
506; top: 47; visibility: hidden" class="company">
<table width="96" border="0" onMouseOver="MM_showHideLayers('sc','','show')"
onMouseOut="MM_showHideLayers('sc','','hide')">
<tr><td class="company" width="90">
<strong>
<font color="#CCCCCC" size="-2">
团购信息</font>
</strong></td></tr>
<tr><td class="company" width="90">
<strong><font color="#CCCCCC" size="-2">
个人购买</font></strong></td></tr>
</table></div>
<div id="fw" style="position:absolute; width:76; height:115; z-index:6; left:
408; top: 49; visibility: hidden">

<table width="96" height="110" border="0" onMouseOver="MM_showHideLayers
('fw','','show')" onMouseOut="MM_showHideLayers('fw','','hide')">
<tr><td class="company"><strong>
<font color="#FF9900" size="1" face="宋体">
<a href="customer.htm">
顾客登记</a></font></strong></td></tr>
<tr><td class="company"><strong>
<font color="#FF9900" size="1" face="宋体">
<a href="prokno.htm">
产品知识</a></font></strong></td></tr>
<tr><td class="company">
<strong><font color="#FF9900" size="1" face="宋体">
<a href="proask.htm">产品咨询</a></font></strong></td></tr>
<tr><td class="company"><strong>
<font color="#FF9900" size="1" face="宋体">
<a href="fix.htm">在线报修</a></font></strong></td></tr>
<tr><td class="company"><strong>
<font color="#FF9900" size="1" face="宋体">
<a href="servicecall.htm">服务热线</a>
</font></strong></td></tr>
```

```
    </table>
  </div></body></html>
```

注意：这个文件代码虽然很长但结构很简单，只是创建了几个表格和其中的标题。然后加上使用功能函数的命令。仔细看看即可明白。

在本代码文件的开头引入了 dht.js 文件和 style.css 文件。

10.1.4 滚动的新闻标题程序文件

1. 程序功能说明

在网站主页上常见滚动的新闻标题，滚动的显示一些国内外的新闻，或发布网站的通知等。滚动的新闻标题程序显示结果如图 10.3 所示。



图 10.3 新闻标题程序显示结果

2. news.htm 程序代码

下面给出滚动新闻程序 news.htm 的源代码，可参考使用。

```
<script language=javascript>
var index=7
link=new Array(6);text=new Array(6);
link[0] ='NBA最新赛事.htm';link[1] ='国际足球赛事.htm';
link[2] ='国内足球赛事.htm';
link[3] ='CBA最新赛事.htm';link[4] ='国际新闻.htm';
link[5] ='国内新闻.htm';link[6] ='乒乓球赛事.htm';
text[0] ='NBA最新赛事';text[1] ='国际足球赛事';
text[2] ='国内足球赛事';text[3] ='CBA最新赛事'
text[4] ='国际新闻';text[5] ='国内新闻';
text[6] ='乒乓球赛事'
document.write ("<marquee scrollamount='1' scrolldelay='100' direction='up' width='150'
height='150'>");
for (i=0;i<index;i++){
```

```
document.write (" <img src='image/1.gif' width='12' height='12'><a href="+link[i]
+" target='_blank'>");
document.write (text[i] +"</A><br>");
}
document.write("</marquee>")
</script>
```

10.1.5 公用的顶部页面程序文件

1. 程序说明

网站大部分页面都可以共用一个顶部内容，它包含网站的标题、标语、导航栏，本程序的功能是创建一个公用的顶部，以便网站的主页与其他页面使用，可以减少重复开发，统一网站风格。

由于顶部内容需要包含导航栏 dht.htm 程序，所以将其页面设计为 jsp 格式，为了使页面整齐可使用表格来布局。在编程前要准备好使用的图片。

公用的页面顶部程序 main-1.jsp 显示结果如图 10.4 所示。



图 10.4 页面顶部程序显示结果

2. main-1.jsp 程序代码

下面给出页面顶部程序 main-1.jsp 的源代码，可参考使用。

```
<head><title>页面顶部</title>
<link rel=stylesheet href="CSS/style.css" type="text/css">
</head>
 <table border="0" align="center" width="934" bgcolor="#000000" style="border-
collapse: collapse">
 <tr><td width="337" height="15" colspan="2"><a href="main.jsp">
<img border="0" src="image/logo_5-2.gif" width="102" height="33"></a>
```

```
<p class="bt"><b>网上体育用品商店</b></font></td>
<td width="649" colspan="6"><font color="#FF0000"><b>
<img border="0" src="image/wel_01.gif" width="669" height="145" align="bottom">
</b></font></td></tr>
<tr bordercolor="#00FF00">
<td width="90"><div align="center"><b><a href="main.jsp">
<font size="3" face="华文楷体">返回主页</font></a></b></div></td>
<td width="90"><div align="center"><font face="华文楷体" size="3"><b><a href="map.htm"
>网站地图</a></font></b></font></div></td>
<td width="90"></td></tr>
</table>
<div class="Layer0" ><%@include file="dht.htm" %></div>
```

说明：为了将导航栏 dht.htm 放在合适位置，可在 style.css 样式表中添加以下代码：

```
.Layer0{
position:absolute; width:160px; height:115px; z-index:11; left: 155px; top: 175px;
}
```

通过<div>标记使用 Layer0 定义的位置确定导航栏在页面上的位置。导航栏是通过 dht.htm 文件直接插入到这个页面中的。

10.1.6 公用的底部页面程序文件

1. 程序功能说明

网站大部分页面都可以共用一个底部内容，它应该包含联系该网站的方式、进入其他网站的友情链接、进入浏览整个网站的网站地图页面、版权说明页面等内容。因为这部分可能要添加到所有页面的下部，所以还应该包括返回主页的方式。本程序功能是可以包含在其他页面与主页中，减少重复开发，统一网站布局。

页面底部程序 main-2.htm 显示结果如图 10.5 所示。

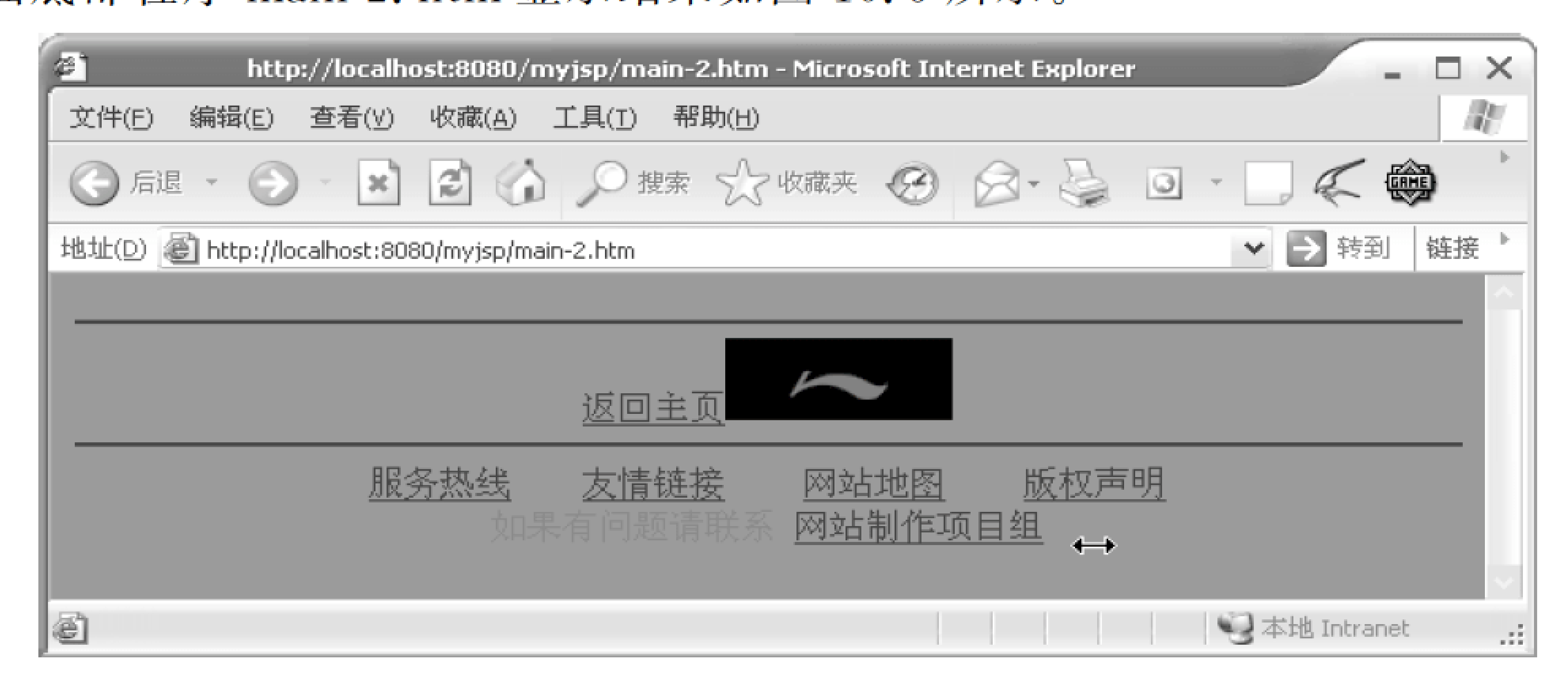


图 10.5 页面底部程序显示结果

2．main-2.htm 程序代码

下面给出页面底部程序 main-2.htm 的源代码，可参考使用。

```
<body bgcolor="#000000">
 <hr color="#0000FF"><div align="center">
 <a href="main.htm">返回主页
<img border="0" src="image/logo_5-2.gif" width="102" height="40">
</a></div>
 <hr color="#0000FF">
 <table width="392" border="0" align="center">
     <tr bordercolor="#00FF00">
        <td width="92" height="15"><div align="center">
<a href="servicecall.htm">服务热线</a></font></div>
</td>
        <td width="93">
<div align="center">
<a href="freiend.htm">友情链接</a></font></div>
</td>
        <td width="97"><div align="center">
<a href="map.htm">网站地图</a></font></div>
</td>
        <td width="92">
<div align="center">
<a href="shming.htm">版权声明</a></font></div>
</td>
     </tr>
   </table>
<div align="center"></div>
<div align="center"><font color="#FF0000">如果有问题请联系</font>
<a href="mailto:zhc@sinc.com">网站制作项目组</a>
</div>
```

10.1.7 公用的数据库连接程序文件

1．程序功能说明

要实现动态功能，经常要用到数据库中的数据。程序 jdbc-roy.jsp 的功能就是实现到连接数据库 roy 的数据源 roy，在以后了解其他数据库时只要修改数据源名称即可。

2．jdbc-roy.jsp 程序代码

下面给出页面底部程序 jdbc-roy.jsp 的源代码，可参考使用。

```
<%@page import="java.sql.*"%>
<%
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection("jdbc:odbc:roy");
Statement s = c. createStatement (ResultSet. TYPE _ SCROLL _ SENSITIVE, ResultSet. CONCUR _
UPDATABLE);
PreparedStatement ps=null;
%>
```

10.1.8 公用的网上搜索引擎

1. 程序功能说明

在网上有许多网站都有搜索引擎的功能。本程序具有使用百度网站搜索引擎进行搜索的功能。

2. 程序代码

使用其他网站搜索引擎的程序文件 bds.html，源代码如下：

```
<form name=query action=http://www1.baidu.com/baidu method=get target=_blank>
<input name=tn type=hidden value=baidu>
<table width="211" border="0">
<tr>
<td colspan="2" class="yonghu"><font color="#CCCCCC">百度搜索</font>
<img src="image/baidulogo.gif" width="80" height="29"></td>
 </tr>
 <tr>
 <td width="132" height="23"><input name="word" type="text" size="18">
 </td>
 <td width="63"><input type="submit" name="Submit3" value="search"></td>
 </tr>
</table>
```

3. 页面运行结果

页面运行结果如图 10.6 所示。

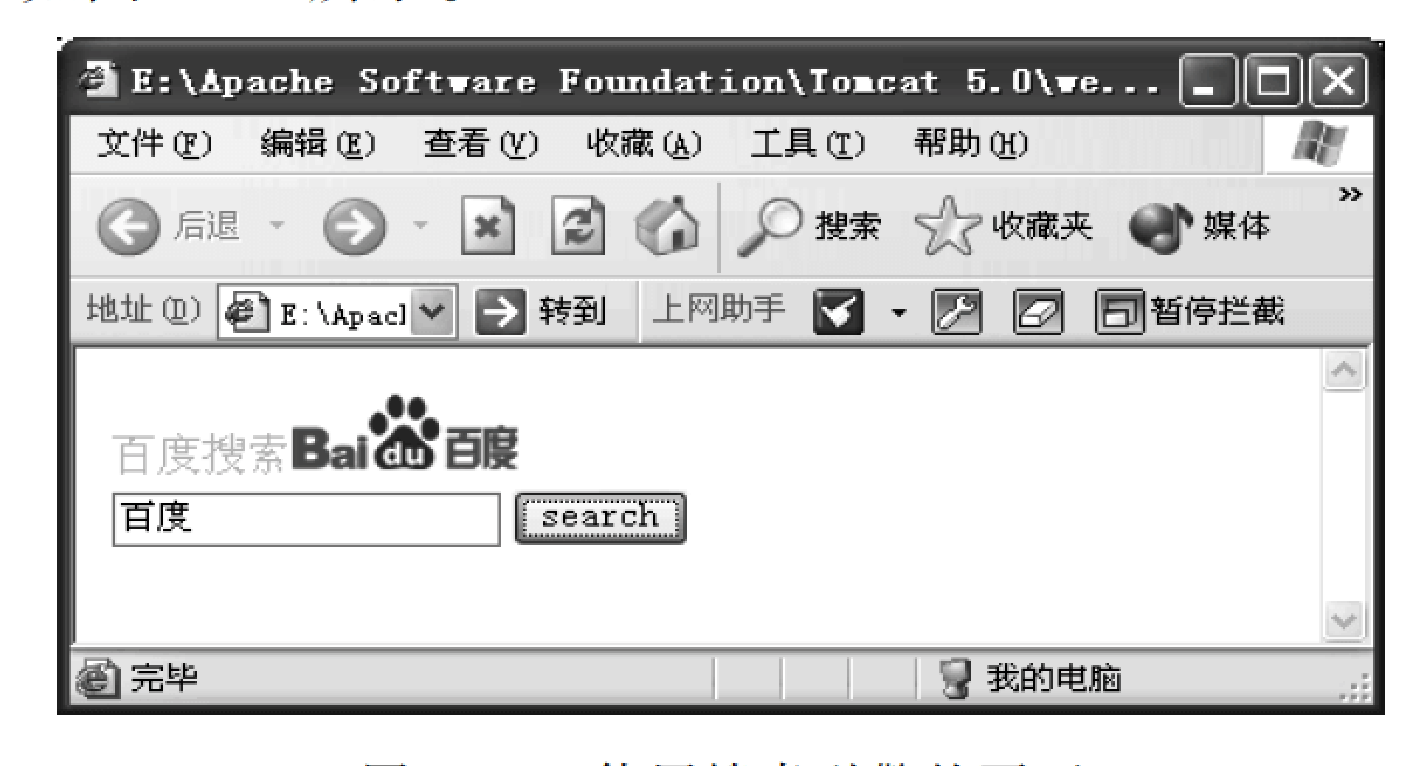


图 10.6 使用搜索引擎的页面

bds.html 文件可以嵌入到其他页面中，提供一个使用百度搜索引擎的接口。

10.2 制作首页

本节主要介绍制作首页的方法。

根据首页的详细设计，该页主要有三部分内容，网站标题、flash 动画、进入主页的图片链接。

制作首页分为如下 2 个环节。

10.2.1 准备首页使用的素材

在制作首页前要准备好该页使用的 flash 动画文件，图片文件素材，要精心制作一个 logo 图片。然后使用编辑器编写首页文件。

10.2.2 编写 index.htm 首页程序文件

1. 程序功能说明

该页面程序功能是通过简洁、漂亮、突出的形式显示网站主题，吸引住浏览用户，提供进入网站主页的接口。

由于该页面没有动态功能要求，所以该页面使用 html 语言来编写，为了使页面整齐可使用表格来布局。

首页在浏览器中的显示结果如图 10.7 所示。

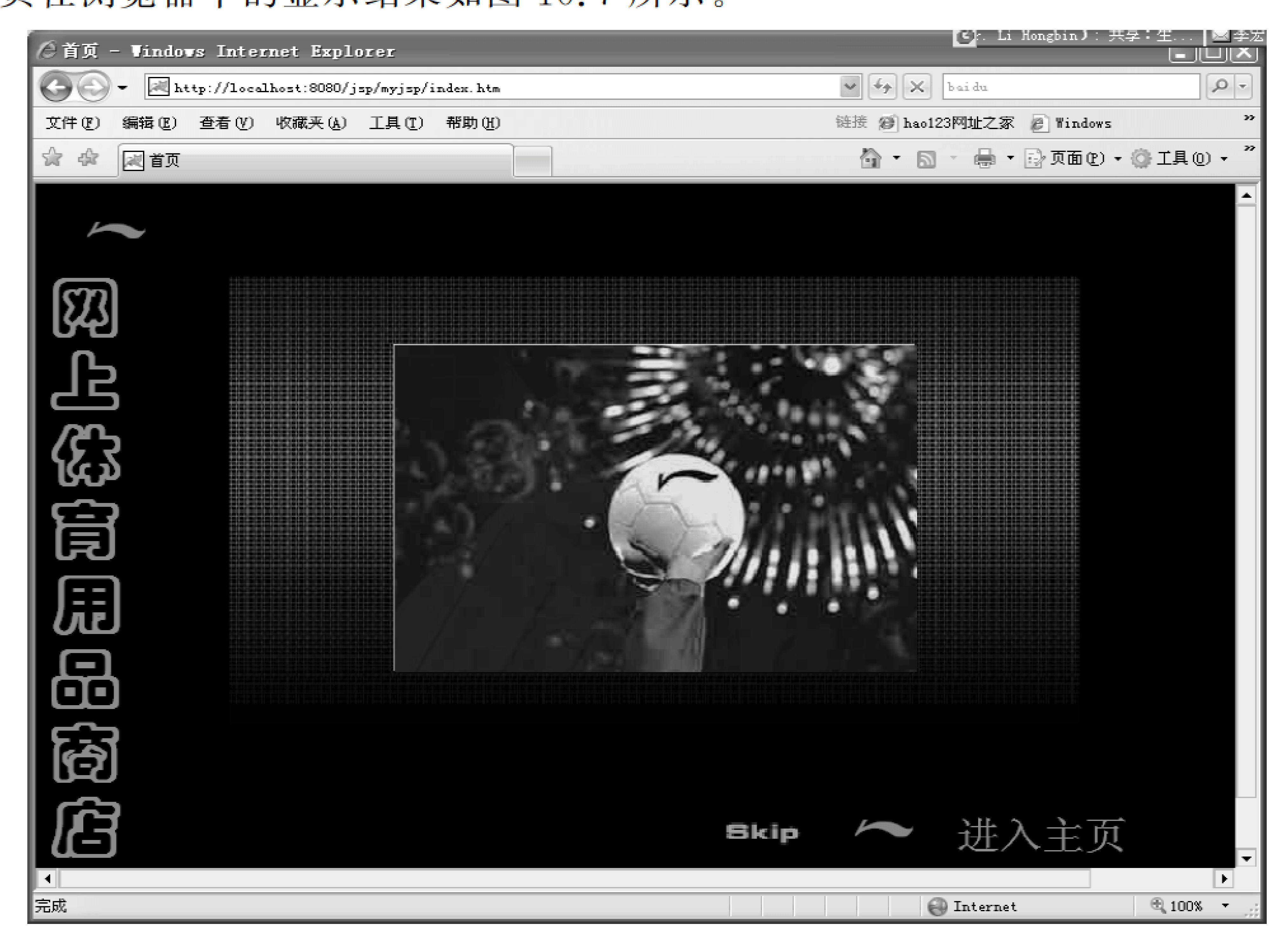


图 10.7 首页页面

2. index. htm 页面程序代码

下面给出首页程序文件 index. htm 的源代码,可参考使用。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>首页</title>
<link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body style="background-color: #000000; background-image: url('')" >
<p><img src="image/logo_5-2.gif" width="102" height="40" border="0"></p>
<table width="990" border="0" cellpadding="0" cellspacing="0" bgcolor="#000000" style=
"border-collapse: collapse" height="518">
<tr>
<td width="93" rowspan="2" valign="top"  height="518">
<p class="bt"><b>网上体育用品商店</b></p>
</td>
<td width="718" height="1"></td>
<td width="48" height="1"></td>
<td width="101" height="1"></td>
</tr>
<tr>
 <td valign="top"  width="718" height="517"><div align="right">
 <embed src="image/lining.swf" quality="high" type="application/x-shockwave-flash"
width="724" height="429">
</embed><a href="main.jsp">
<img src="image/skipfinal.gif" width="101" height="25" border="0"></a>
<img src="image/logo_5-2.gif" width="102" height="40" border="0">
<a href="main.jsp">
<font size="6">进入主页</font></a></div></td>
<td height="517" width="149" colspan="2"></td>
</tr>
 </table>
</body></html>
```

10.3 制作会员登录和注册动态功能模块

现在大多数的网站都实行会员制,以区别一般浏览者与会员,会员用户将提供更多的优惠政策。实现会员制需要提供会员注册和登录的功能,通过注册将用户信息保存在服务器的数据库中,标记其为会员。注册过的老会员可以登录到该网站,将享受网站提供的更好的服务。

本节的内容主要介绍制作创建会员登录和注册动态功能模块的方法，主要介绍用来实现用户注册与登录、实现会员制的 5 个程序文件，各个应用程序内容如下所述。

10.3.1 会员登录和注册入口程序文件

1. 程序功能说明

login.htm 页面程序的功能是提供一个会员登录和注册入口，如果是注册过的老会员，可以在这里输入会员名和密码，单击“登录”按钮将填写的会员数据提交给登录处理程序 login.jsp 判别其会员身份。如果是新用户要注册，单击“新会员注册入口”链接将进入登录页面 register.htm。其页面如图 10.8 所示。



图 10.8　会员登录和注册入口页面

2. login.htm 程序代码

下面给出登录和注册入口页面程序文件 login.htm 的源代码，可参考使用。

```
<title>会员登录入口</title>
<body background="image/di.gif" link="#CCCCCC" vlink="#666666">
<table width="211" border="0" bgcolor="#000000">
<tr><td height="22" colspan="2" valign="top" class="yonghu" width="205">
<p align="center"><font color="#CCCCCC">会员登录</font></td></tr>
<tr><td height="22" colspan="2" valign="top" width="205">
 <form name="form1" method="post" action="login.jsp">
<span class="yonghu"><font color="#CCCCCC">会员名</font></span>
<input name="username" type="text" id="username" size="15"></td></tr>
<tr><td height="20" colspan="2" valign="top" width="205">
<span class="yonghu"><font color="#CCCCCC">密   码</font></span>
<input name="password" type="password" id="password" size="15">
</td></tr>
<tr><td width="53" height="20"><div align="center">
<input type="submit" name="Submit" value="登录">
</form>
```

```
</div></td>
<td width="148"  class="yonghu">
<a href="register.htm" >新会员注册入口</a></td></tr>
</table>
```

10.3.2 用户登录数据处理应用程序文件

1. 程序功能说明

在登录和注册入口页面 login.htm 中用户输入会员名 yhm 与口令 mm 后，单击“登录”按钮后，会员输入的数据将提交给 login.jsp 处理程序，该程序的功能是将输入的数据传递到数据库，按输入的会员名查找是否有该会员名及口令，如果口令相符，将打开一个欢迎会员进入的页面 user.jsp。

login.jsp 处理程序没有页面只是提供数据处理功能。

2. login.jsp 程序代码

下面给出会员用户登录数据处理应用程序 Login.jsp 的源代码，可参考使用。

```
<%@page contentType="text/html; charset=gb2312"%>
<%@page import="java.sql.*"%>
<%@include file="jdbc-roy.jsp" %>
<html>
<body>
<b>在客户端浏览器向数据库添加数据</b><p>

<%
  //得到从浏览器传来的用户名和密码的值
String yhm=request.getParameter("yhm");
String mm=request.getParameter("mm");
  //在数据库中按输入的会员名查找口令
String sql2="SELECT mm FROM userpwd WHERE yhm='"+yhm+"'";
ResultSet rs=s.executeQuery(sql2);
//判断口令是否正确
String pass=null;
 while(rs.next()){  pass=rs.getString(1); }
 if(mm.equals(pass)){
 session.setAttribute("username",yhm);
%>
<jsp:forward  page="user.jsp"/>
 <%
 }else{
    out.print("登录失败!用户名不存在或密码不正确。可返回主页重新登录或先进行注册再登
录。<a href=main.jsp>返回主页</a>");
    }
```

```
rs.close();
ps.close();
c.close();
%>
</body>
</html>
```

10.3.3 欢迎会员登录成功的页面文件

1. 程序功能说明

欢迎会员登录的页面 user.jsp 给出会员成功登录的信息,提供一些会员可以进入的链接。

2. user.jsp 页面程序代码

下面给出会员登录页面程序文件 user.jsp 的源代码,可参考使用。

```
<head>
<title>会员登录页面</title>
<%@page contentType="text/html;charset=ISO8859_1" %>
<%@page import="java.util.* " %>
<%
//保证此页面一定要通过用户登录才能进入
if(session.getAttribute("username")==null){
 response.sendRedirect("login.htm");
}
else
{
 String username=(String)session.getAttribute("username");
%>

<link href="css/style.css" rel="stylesheet" type="text/css">
</head>
<body bgcolor="#FFFFFF" background="image/di.gif" >
<%@include file="main-1.jsp" %>
<%
Date current=new Date();
int  time=current.getHours();
%>
<div class="Layer2" >
  <p class="customer"><%=username %>
  <%if (time<=12&&time>=6)
            out.println("上午好!");
          else if(time<=18)
             out.println("下午好!");
```

```
            else   out.println("晚上好!");
%>
</p>
</div>
<div class="Layer3">
<p class="customer">尊敬的用户,您已经登录成功!这里是会员天地,你可以在这里享受特殊的服务祝你在这里开心!</p>
<p class="customer" align="center">
<a href="chat/index.jsp">进入聊天室</a></font></p>
<p align="center"><a href="lyb/index.jsp">进入留言板</a></p>
<p align="center"><a href="tp/showquestion.jsp">体育明星投票</a></p>
<p align="center" class="customer"><u><a href="main.jsp">返回主页</a></u></p>
</div>
<%}%>
</body>
</html>
```

其页面如图 10.9 所示。

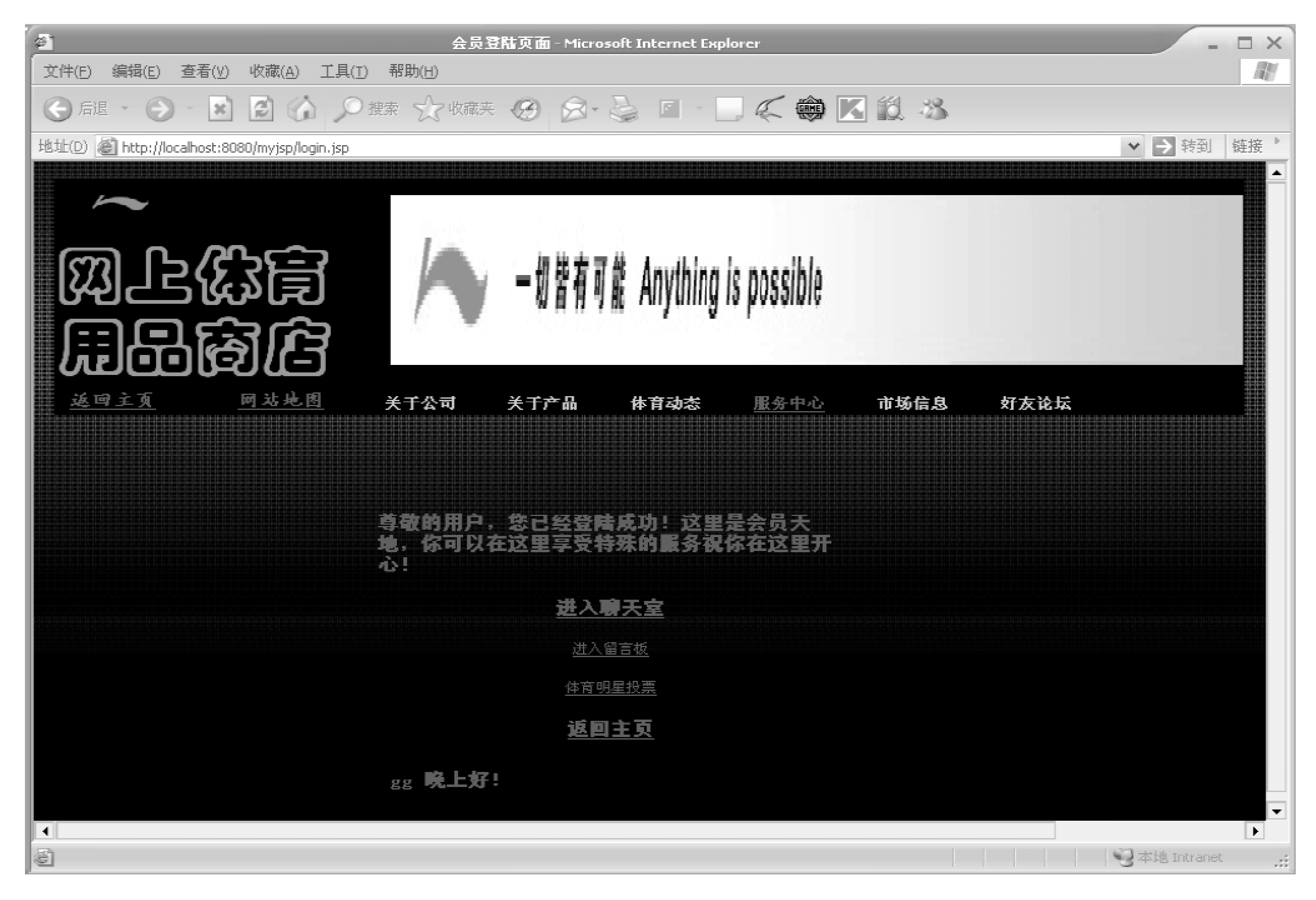


图 10.9　会员注册登记页面

10.3.4　会员注册页面程序文件

1. 程序功能说明

本页面是在“登录和注册入口”单击“新会员注册入口”超链接后打开的页面，本页面的功能是输入新会员的信息，并按“提交”按钮将登记内容提交给 JSP 程序 register.jsp 进

行处理。会员注册的页面 register.htm 如图 10.10 所示。



图 10.10　会员注册登记页面

2. register.htm 程序代码

下面给出会员注册登记页面程序文件 register.htm 的源代码，可参考使用。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>会员注册</title>
<link href="css/style.css" rel="stylesheet" type="text/css">
</head>

<body background="image/di.gif" link="#CCCCCC" vlink="#666666">
  <p class="bt" align="center">会员注册</p>
  <p class="page">欢迎您来到我们的网站！在此您可以通过免费注册成为我公司的会员，作为
我公司的会员，会定期收到我们给你发去的产品资料。让你更快地了解我们的信息。通过以下资
料的填写你就可以成为我的会员了，快快行动吧！</p>
  <form action="register.jsp" method="post" name="form1" class="page">
    <p class="page" align="left">会员账号：
      <input name="yhm" type="text" id="yhm" size="15">
      (6 位以上的数字或英文字母，必填)</p>
    <p class="page" align="left">密   码 (必填)：
      <input name="mm" type="password" id="mm" size="15">确认密码：
      <input name="password2" type="password" id="password2" size="15"></p>
```

```
    <p class="page" align="left">姓     名:
      <input name="name" type="text" id="name" size="15">联系电话:
      <input name="call" type="text" id="call" size="15">
    </p>
    <p class="page" align="left">通信地址:
      <input name="address" type="text" id="address" size="50">
    </p>
    <p class="page" align="left">E-mail(必填):  
      <input name="email" type="text" id="email" size="50">
    </p>
    <p align="center" class="page">
      <input type="submit" name="Submit" value="提交">
          
      <input type="reset" name="Submit2" value="重置">
    </p>
  </form>
  </body>
</html>
```

10.3.5 注册数据保存处理程序文件

1. 程序功能说明

在会员注册登记页面 register.htm 中登记的数据信息按“提交”按钮后交给 register.jsp 处理程序,该程序的功能是将登记数据传递到数据库中。保存数据到数据库后即注册成功,将打开一个成功的页面 succ.htm。

register.jsp 处理程序没有页面只是提供数据处理功能。

2. register.jsp 程序代码

下面给出注册数据保存处理程序文件 register.jsp 的源代码,可参考使用。

```
<%@page contentType="text/html; charset=gb2312"%>
<%@page import="java.sql.*"%>
<%@include file="jdbc-roy.jsp"%>

<html>
<body>
<b>在客户端浏览器向数据库添加数据</b><p>

<%
//得到从浏览器传来的用户名和密码的值
String yhm=request.getParameter("yhm");
String mm=request.getParameter("mm");
String  name1=request.getParameter("name");
```

```
//姓名为中文的处理
String  name=new String (name1.getBytes("ISO-8859-1"),"GBK");
String  call=request.getParameter("call");
String  address1=request.getParameter("address");
String  address=new String (address1.getBytes("ISO-8859-1"),"GBK");
String  email=request.getParameter("email");

String sql1="insert into userpwd(yhm,mm) values('"+yhm+"','"+mm+"')";
s.executeUpdate(sql1);

String sql2="insert into register(yhm,name,call,address,email) values('"+yhm+"','"+name
+"','"+call+"','"+address+"','"+email+"')";
s.executeUpdate(sql2);

%>
<jsp:forward  page="succ.htm" />
<%
ps.close();
s.close();
c.close();
%>
</body>
</html>
```

注意：通过 register.jsp 处理程序可将输入的会员信息数据添加到数据库 roy 中。

如果注册成功，可以看到如图 10.11 所示的页面(succ.htm，该页面只有一句话和用户返回主页的链接，这里就不介绍了)。



图 10.11　会员注册成功页面

在看程序时注意,有的程序语句被分为了两行,这是排版造成的,在编写程序时不能将一个程序语句写为两行。例如下面两行为一个程序语句:

```
String sql2="insert into register(yhm,name,call,address,email) values('"+yhm+"','"+name
+"','"+call+"','"+address+"','"+email+"')";
```

10.4　整合主页

本节主要介绍制作模块化主页的方法。

10.4.1　主页的构成

这里介绍的主页是由上面介绍的多个页面程序文件整合构成的,包括共用的程序、动态功能模块应用程序等,主页程序文件只是通过 JSP 语言将这些程序文件组合在一起,确定他们在主页上的具体位置,主页页面如图 10.12 所示。



图 10.12　"网上体育用品商店"网站主页页面

10.4.2　主页程序文件 main.jsp 的代码

主页程序文件 main.jsp 的源代码如下所示,可参考使用。

```
<%@page contentType="text/html;charset=ISO8859_1" %>
<link href="css/style.css" rel="stylesheet" type="text/css">
<html>
<body>
<%@include file="main-1.jsp" %>
    <%@include file="login.htm" %>
    <h2><a href="tp/admin/login.jsp" class="zhuye">管理员入口</a></h2>
<body bgcolor="#FFFFFF" background="image/di.gif" >
<div class="layer1"><%@include file="news.htm" %></div>
<div class="layer2">
<%@include file="jsq2.jsp" %>
</div>
<div class="layer4"><img src="image/xyt2.gif" width="400" height="260">
</div>
<div class="layer5"><%@include file="main-2.htm" %></div>
</body>
</html>
```

注意：制作主页的任务还没有全部完成，其中"管理员入口"是给进行市场调查动态模块留下的接口，滚动新闻标题对应的静态内容页面也没有完成，各个栏目页面没有完成，这些都要留到下个阶段来完成。

主页程序看起来很简明。因为它是由其他编写好的程序文件组合构成的，通过使用多个<div class="Layer5">标记确定每个模块或其他图片、接口在页面上显示的位置。这样编写程序，可提高模块化，而且便于维护和修改，并有利于代码重用。这样的程序结构称为模块化的程序结构。

10.5 制作栏目页面及相关二级页面

本节的内容主要介绍制作网站各个栏目一级页面与相关二级页面的注意问题。

10.5.1 制作栏目一级页面

栏目一级页面由多个页面组成，具体个数由栏目多少来确定。栏目页面要包括：栏目页面导航栏、菜单、栏目主要内容介绍、到二级页面的链接、返回主页的链接、到其他栏目页面的链接等。

图 10.13 为"关于公司"栏目的一级页面，图 10.14 为"关于产品"栏目的一级页面，可参考开发主页的步骤开发各个栏目的一级页面。

10.5.2 制作相关二级页面

相关二级页面包括两部分内容：主页相关的二级页面与栏目页面相关的二级页面。



图 10.13 “关于公司”栏目一级页面



图 10.14 “关于产品”栏目一级页面

1. 主页相关的二级页面

要由专人负责主页的制作，包括实现与主页相关的二级页面。例如“网上体育用品商店”主页中滚动新闻标题超链接对应各个静态页面，只要给出最新的文字体育信息与图片。

可以使用数据库来存放新闻内容。

2. 栏目页面相关的二级页面

二级页面主要包括：显示主页菜单中指出的文字内容介绍、产品展示(图片)、与用户进行交互、保存用户输入的数据等、返回栏目页面的超链接等。本任务功能简单，但工作量大。如果前期素材收集工作准备充分，将很容易完成。

图 10.15 为“关于产品”栏目一级页面中“器材配件”菜单的二级页面。



图 10.15　“关于产品”一级页面的二级页面

10.6　创建用户留言动态功能模块

用户留言系统是一般网站都要具备的，它是了解用户意见的一个主要窗口。

本节的内容主要介绍如何编写一个用户留言系统的应用程序，通过创建用户留言系统，可以了解如何保存用户的留言信息，如何显示留言信息，如何在页面的表格中显示图标图片，如何按留言的时间对信息排序显示，如何在页面中显示 IP 地址、E-mail 地址、QQ 号码等。

10.6.1　共用的 CSS 程序文件

1．程序功能说明

lyb.css 样式表程序文件的功能是统一确定留言相关页面的背景、主题色彩、文字属性、链接、位置、颜色等内容。

2．程序代码

lyb.css 的源代码如下所示，可参考使用。

```
.bg {
    background-color: #000000;
    list-style-type: square;
```

```
}
body {background-color:#eff3ff;font-size: 12px;
scrollbar-3dlight-color: #4f81ca;
scrollbar-arrow-color: #ffffff;
scrollbar-track-color: #4f81ca;
scrollbar-base-color: #4f81ca;
}
td {
    font-size: 9pt; line-height: 135%; font-family: "宋体", "verdana", "arial", "
helvetica", "sans-serif"
}
a {
    font-size: 9pt; line-height: 135%; font-family: "宋体", "verdana", "arial", "
helvetica", "sans-serif"
}
input {
    background-color: #ffffff; border-bottom: #4f81ca 1px solid; border-left: #4f81ca
1px solid; border-right: #4f81ca 1px solid; border-top: #4f81ca 1px solid; font-size:
9pt;color:#000080
}
select {
    font-size: 9pt; font-family: "宋体", "verdana", "arial", "helvetica", "sans-serif"
}
textarea {
    font-size: 9pt; font-family: "宋体", "verdana", "arial", "helvetica", "sans-serif"
}
option {
    font-size: 9pt; font-family: "宋体", "verdana", "arial", "helvetica", "sans-serif"
}

a:link {
    color: #003303; text-decoration: none
}
a:hover {
    color: #0066ff; text-decoration: underline
}
a:visited {
    text-decoration: none; color: #003303
}
```

10.6.2 共用的 JavaBean 程序文件

1. JavaBean 程序功能

留言系统将使用两个 JavaBean 应用程序：Bean.java 与 PageCt.java。

Bean.java 程序的功能之一是通过 ex_chinese(string str)方法定义给定的中文输出格式为 gb2312(简体中文),功能之二是通过 gettime()方法得到当前时间,其中使用了得到当前时间的 java.util.Date()类。

PageCt.java 程序的功能是定义 JSP 页面一页显示 10 条记录、得到当前页数、下一页、上一页、总记录数、总页数等方法。

2. JavaBean 源代码

bean.java 源程序代码如下所示,可参考使用。

```
package lyb;
import java.text.* ;
public class bean {

  public bean() {   }

  public String ex_chinese(String str){
     if(str==null){ str  ="" ; }
     else{
         try {
        str=new String(str.getBytes("iso-8859-1"),"gb2312") ;
         }
         catch (Exception ex) {    }
     }
     return str ;
  }

public String gettime() {
     String strCurrentDate="";
     try {
     java.util.Date date=new java.util.Date();
     SimpleDateFormat strDate=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
     strCurrentDate=strDate.format(date);
     }
     catch (Exception ex) {     }
     return strCurrentDate ;
  }
}
```

PageCt.java 源程序代码如下所示,可参考使用。

```
package lyb;
public class PageCt{
     private long l_start;              //开始记录
     private long l_end;                //结束记录
```

```
    private long l_curpage;          //当前页数
    private long l_totalnum;         //总记录数
    private int int_num=10;          //每页 10 条
    private long l_totalpage;        //总的页数

    public void Init(long currentpage,long totalnum)  {
        l_curpage=currentpage;
        l_totalnum=totalnum;

        if (currentpage>=0)  {
            if (currentpage>=(long)Math.ceil((double)l_totalnum/(double)int_num))
              l_curpage=(long)Math.floor((double)l_totalnum/(double)int_num);
            else
               l_curpage=currentpage;
        }
        else{ l_curpage=0;  }
        l_start=l_curpage * int_num;
        l_end   =l_start +int_num;
        if (l_end >l_totalnum)
        l_end=l_totalnum;
        l_totalpage=(long)Math.ceil((double)l_totalnum/(double)int_num);
    }

    public long getCurpage()  {return l_curpage;}
    public long getPrepage()  {
           if (l_curpage-1>=0)  {return l_curpage-1;}
         else {return 0;}
    }
    public long getNextpage()  {
        if (l_curpage+1<=l_totalpage)  {return l_curpage+1;}
        else{return l_totalpage;}
    }
    public long getTotalnum()  {return l_totalnum;}
    public long getTotalpage()  {return l_totalpage;}
    public long getStart()  {return l_start;}
    public long getEnd()  {return l_end; }
}
```

注意：

(1) 要将 JavaBean 文件编译为 class 字节码文件。

(2) 要注意存放 JavaBean 的位置，一般放在 Tomcat 的根目录/ROOT 下，例如，可将 bean.class、PageCt.class 文件存放在 C:/Tomcat/webapps/ROOT/WEB-INF/classes/lyb/文件夹中。如果已经设置好虚拟目录，可存放在虚拟目录下的 WEB-INF/classes/路径下，例如，C:/wzkf/WEB-INF/classes/lyb 文件夹中，要注意 classes/下子文件夹的名

字要与程序中定义的类包名字相同。

10.6.3 共用的 JavaScript 功能程序文件

1. week.js 程序功能

week.js 程序的功能是可以将代码插入到任意页面文件中，在页面上显示当前年、月、日、星期几的数据信息。

2. week.js 程序代码

week.js 程序的源代码如下所示，可参考使用。

```
var day="";
var month="";
var ampm="";
var ampmhour="";
var myweekday="";
var year="";
mydate=new Date();
myweekday=mydate.getDay();
mymonth=mydate.getMonth()+1;
myday=mydate.getDate();
myyear=mydate.getYear();
year=(myyear>200)? myyear:1900+myyear;
if(myweekday ==0)
weekday=" 星期天 ";
else if(myweekday ==1)
weekday=" 星期一";
else if(myweekday ==2)
weekday=" 星期二 ";
else if(myweekday ==3)
weekday=" 星期三";
else if(myweekday ==4)
weekday=" 星期四 ";
else if(myweekday ==5)
weekday=" 星期五 ";
else if(myweekday ==6)
weekday=" 星期六 ";
document.write(year+"-"+mymonth+"-"+myday+weekday);
```

10.6.4 questionnaire 数据库与 lyb 表

1. lyb 表的作用

lyb 表的作用是存放用户的留言。

2. lyb 的结构

lyb 可以单独创建一个数据库，为了简化，可将 lyb 表存放在已经创建的 Access 数据库 questionnaire 中。lyb 的物理结构如图 10.16 所示。



图 10.16　lyb 的物理结构

10.6.5　连接数据库的 JSP 程序文件

1. opendata.jsp 程序功能说明

opendata.jsp 程序的功能是连接 questionnaire 数据库文件，为 JSP 文件使用数据库中的数据铺平道路。

2. opendata.jsp 程序代码

连接数据库程序文件 opendata.jsp 的源代码如下所示，可参考使用。

```
<%@page import="java.sql.*"%>
<%
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String url="jdbc:odbc:jsp";
Connection c=DriverManager.getConnection(url);
Statement s=(Statement)c.createStatement();
PreparedStatement ps;
ResultSet rs;
String sql;
%>
```

10.6.6 用户留言模块主页

1. lyb-index.jsp 页面程序功能

用户留言主页是进入用户留言模块看到的页面，主要用来显示用户的留言。留言页面按留言的时间降序显示留言的内容，在页面下方可以看到留言的总页数、当前页数，输入页数可以查找该页留言，按“上页”/“下页”按钮可以翻页，每个页面只显示 10 条留言记录。

主页提供了用户发布留言的接口“我要留言”超链接，单击这个超链接可进入用户留言发布页面。

2. lyb-index.jsp 页面程序功能程序代码

留言主页 lyb-index.jsp 的源代码如下所示，可参考使用。

```
<%@page contentType="text/html; charset=gb2312" language="java" import="java.sql.*"
errorPage="" %>
<%@include file="opendata.jsp"%>

<html><head><title>用户留言模块 Version1.0</title>
<link href="lyb.css" rel="stylesheet" type="text/css">
<jsp:useBean id="PageCt" scope="page" class="lyb.PageCt" />
</head>

<body>
<TABLE cellSpacing=1 cellPadding=3 width=98% align=center background="images/top_line.
gif" border=0>
  <TR align=middle>
    <TD width="20%" align="left" nowrap><FONT color=#78b3f9>  
      <script src="week.js" language="JavaScript" type="text/javascript">
      </script>
      </FONT></TD>
    <TD width="20%" nowrap>
    <div id="Layer1" style="position:absolute; width:198px; height:62px; z-index:1;
left: 284px; top: 31px;">
    <img src="../image/logo6gif.gif" width="160" height="35" border="0"></div></TD>
    <TD width="20%" nowrap></TD>
    <TD width="20%" nowrap> </TD>
    <TD width="20%" nowrap></TD>
  </TR>
</TABLE>
<TABLE height=76 cellSpacing=0 cellPadding=0 width=98% align=center background=
"images/bottom_line.jpg" border=0>
  <TR>
```

```
    <TD>
<TABLE height = 72 cellSpacing = 0 cellPadding = 0 width =" 99% " align = center bgColor =
#ffffff border=0>
          <TR>
            <TD width="163" align="center">
            <IMG src="images/login.gif" border=0 width="154" height="60"></TD>
            <TD width="476" align=right></TD>
            <TD align=center width=123>
<TABLE width="90% " border=0 cellPadding=1 cellSpacing=0>
                <TR vAlign=center align=middle>
                  <TD width="30% ">
<IMG height=16  src="images/homepage.gif" width=16></TD>
                  <TD width="70% "><a href="../main.jsp">网站主页</a></TD>
                </TR>
                <TR vAlign=center align=middle>
                  <TD></TD>
                  <TD></TD>
                </TR>
                <TR vAlign=center align=middle>
                  <TD><IMG height=16
                    src="images/Favorites.gif"
                    width=16></TD>
                  <TD><a href="../user.jsp">用户天地</a></TD>
                </TR>
              </TABLE></TD>
          </TR>
        </TABLE></TD>
  </TR>
</TABLE>
<table width="98% " align="center" cellpadding="0" cellspacing="0" background="images/
index-t.gif">
  <tr>
    <td width="5% " align="left">

    </td>
    <td height="22" align="left"><a href="say.jsp">我要留言</a></td>
  </tr>
</table>

<table width="98% " border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td height="3"></td>
  </tr>
</table>
```

```
<table width="98%" border="0" align="center" cellpadding="0" cellspacing="0" bgcolor="#4f81ca">
  <tr>
    <td><table width="100%" border="0" align="center" cellpadding="1" cellspacing="1">

<%
//查找数据库中的留言记录数
ResultSet rs0=s.executeQuery("select count(*) from lyb");
rs0.next();
long data_num=rs0.getLong(1);
long Current_Page=0;
String currentpage=(String)request.getParameter("currentpage");
if (currentpage!=null &&!currentpage.equals(""))
{
  Current_Page=Integer.parseInt(request.getParameter("currentpage"));
}
String Query_Page=(String)request.getParameter("Query_Page");
if (Query_Page!=null &&!Query_Page.equals(""))
{
  Current_Page=Integer.parseInt(request.getParameter("Query_Page"))-1;
}
PageCt.Init(Current_Page,data_num);
long l_start=PageCt.getStart();
long l_end=PageCt.getEnd();

//查询记录
rs=s.executeQuery("select * from lyb order by id desc");
long i=0;
while((i<l_start) && rs.next()){ i++;
}
//输出查询结果
long j=0;
while(rs.next() && (i<l_end))
{  j=i+1;
%>
        <tr
<%if((i%2)!=0){%>bgcolor="#78b3f9"<%}else{%>bgcolor="#BEDAFC"<%}%>>
          <td width="20%" align="center" >留言人:<%=rs.getString("name")%>
<img src="<%=rs.getString("sex")%>" border="0"></td>
          <td><table width="100%" border="0" cellpadding="0" cellspacing="0">
             <tr align="left"<%if((i%2)!=0){%>bgcolor="#78b3f9"<%}else{%>bgcolor
="#BEDAFC"<%}%>>
                <td width="25%"><a href="<%=rs.getString("www")%>">
```

```
                    <img src="images/HOME.gif" alt="个人主页" width="16" height=
"16" border="0" align="absmiddle">
                    </a>个人主页</td>
                    <td width="25%">
<img src="images/ip.gif" alt="" width="13" height="15" align="absmiddle">
                    IP: <%=rs.getString("IP")%></td>
                    <td width="25%"><a href="mailto:<%=rs.getString("email")%>">
                    <img src="images/EMAIL.gif" alt="留言人 E-Mail" width="16" height="16"
border="0" align="absmiddle">
                    </a>E-Mail</td>
                    <td width="25%">
    <img src="images/oicq.gif" alt="" width="16" height="16" align="absmiddle">
                    QQ:<%=rs.getString("oicq")%></td>
                  </tr>
                </table></td>
            </tr>
            <tr
<%if((i%2)!=0){%>bgcolor="#78b3f9"<%}else{%>bgcolor="#BEDAFC"<%}%>>
              <td width="20%" align="center">
    <img src="<%=rs.getString("face")%>" border="0"></td>
              <td><%=rs.getString("memo")%></td>
            </tr>
            <tr
<%if((i%2)!=0){%>bgcolor="#78b3f9"<%}else{%>bgcolor="#BEDAFC"<%}%>>
              <td width="20%" align="center">时间：<%=rs.getString("addTime")%></td>
              <td></td>
            </tr>
            <tr>
              <td colspan="2" align="center" bgcolor="#ffffff">
<hr size="1" color="#000080" width="100%"></td>
            </tr>
            <%
    i++;
}%>
            <tr align="right" valign="top">
              <form method=GET action=lyb-index.jsp>
                <td colspan="2" bgcolor="#FFFFFF">共<%=PageCt.getTotalnum()%>条<%=
PageCt.getCurpage()+1%>/<%=PageCt.getTotalpage()%>
                  页 查看第
                  <input type=text name=Query_Page size=3>
                  页<a href=lyb-index.jsp?currentpage=<%=PageCt.getPrepage()%>>上页</a>
<a href=lyb-index.jsp?currentpage=<%=PageCt.getNextpage()%>>下页</a>
                     </td>
              </form>
```

```
        </tr>
      </table></td>
  </tr>
</table>
<table width="98%" border="0" align="center" cellpadding="2" cellspacing="2"
background="images/bottom_line.gif">
  <tr>
    <td width="27%"></td>
    <td width="73%"></td>
  </tr>
  <tr>
    <td></td>
    <td>建议 800*600 IE5.5以上(推荐6.0)
</td>
  </tr>
  <tr>
    <td></td>
    <td></td>
  </tr>
</table>
</body>
</html>
```

3. 页面程序运行结果

用户留言主页 lyb-index.jsp 页面的运行结果如图 10.17 所示。



图 10.17　用户留言主页

注意：

(1) 本程序文件虽然很长，主要是表格的各种标记，要注意在表格中使用 JSP 语句的方式。

(2) 本程序有多个程序语句被分为了两行,这是排版造成的,在编写程序时要注意不能将一个程序语句写为两行。

10.6.7 发布用户留言页面

1. say.jsp 页面程序功能

在顾客留言系统主页单击“我要留言”链接可进入 say.jsp 发布留言页面,该页面的功能是让用户输入留言内容,输入或选择用户的信息。

2. say.jsp 页面程序代码

下面给出用户留言主页 say.jsp 的源代码,可参考使用。

```
<html>
<link href="lyb.css" rel="stylesheet" type="text/css">
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<table height=76 cellSpacing=0 cellPadding=0 width=821 align=center background=
"images/bottom_line.jpg" border=0>
  <tr>
    <td width="821">
    <table height=72 cellSpacing=0 cellPadding=0 width="99%" align=center bgColor=#
ffffff border=0>
        <tr>
          <td width="163" align="center">
          <img src="images/login.gif" border=0 width="154" height="60">
          </td>
          <td width="476" align=right></td>
          <td align=center width=123>
          <table width="90%" border=0 cellPadding=1 cellSpacing=0>
              <tr vAlign=center align=middle>
                <td><img height=16 src="images/homepage.gif" width=16></td>
                <td><a href="../main.jsp">  网站主页</a></td>
              </tr>
              <tr vAlign=center align=middle><td></td><td></td></tr>
              <tr vAlign=center align=middle>
                <td><img height=16 src="images/Favorites.gif" width=16></td>
                <td> <a href="../user.jsp"> 会员天地主页</a></td>
              </tr>
            </table></td>
        </tr>
      </table></td>
  </tr>
</table>
<table width="820" align="center" cellpadding="0" cellspacing="0" background="images/
```

```
index-t.gif">
  <tr><td width="45" align="left"></td>
    <td height="22" align="left" width="775"><b>
    <a href="lyb-index.jsp">去看留言</a>
    </b></td>
  </tr>
</table>
<table width="815" border="0" align="center" cellpadding="1" cellspacing="1"
bordercolor="4f81ca" bgcolor="4f81ca">
  <tr>
    <td width="811">
    <table width="100%" border="0" align="center" cellpadding="1" cellspacing="1">
        <form name="form" action="saysucc.jsp" method="post">
          <tr bgcolor="#FFFFFF">
            <td width="40%" align="right">用户名:</td>
            <td width="50%"><input type="text" name="name" size="20"></td>
          </tr>
          <tr bgcolor="#FFFFFF">
            <td align="right">性别:</td>
            <td><input type=radio CHECKED value=images/Male.gif name=sex>
  <img  src=images/Male.gif width="23" height="21">男孩     
  <input type=radio value=images/Female.gif name=sex>
  <img  src=images/Female.gif width="23" height="21"><font>女孩</font></td>
          </tr>
          <tr bgcolor="#FFFFFF">
            <td width="40%" align="right">个人主页地址:</td>
            <td width="50%"><input type="text" name="www" size="20"></td>
          </tr>
          <tr bgcolor="#FFFFFF">
            <td width="40%" align="right">电子邮件地址:</td>
            <td width="50%"><input type="text" name="email" size="20"></td>
          </tr>
          <tr bgcolor="#FFFFFF"><td align="right">你的 OICQ 号:</td>
            <td><input type="text" name="oicq" size="20"></td>
          </tr>
          <tr bgcolor="#FFFFFF">
            <td align="right">选择留言使用的头像:</td>
            <td><select name=face size=1 onChange="document.images['face'].src=options
[selectedIndex].value;" style="BACKGROUND-COLOR: #99CCFF;COLOR: #000000">
                <option value="images/Image1.gif" selected>image1.gif</option>
                <option value="images/image2.gif">image2.gif</option>
                <option value="images/image3.gif">image3.gif</option>
                <option value="images/image4.gif">image4.gif</option>
                <option value="images/image5.gif">image5.gif</option>
```

```
            <option value="images/image6.gif">image6.gif</option>
            <option value="images/image7.gif">image7.gif</option>
            <option value="images/image8.gif">image8.gif</option>
            <option value="images/image9.gif">image9.gif</option>
            <option value="images/image10.gif">image10.gif</option>
             </select> 
             <img src=images/Image1.gif width="32" height="32" id=face> 
            选择头像
        </td></tr>
      <tr bgcolor="#FFFFFF"><td align="right">在这里写留言内容：</td>
        <td><textarea name="memo" cols="50" rows="10"></textarea></td>
      </tr>
      <tr bgcolor="#FFFFFF">
        <td align="right"><input type="submit" name="Submit" value="提交"></td>
        <td><input type="reset" name="Submit2" value="重置"></td>
      </tr>
    </form>
  </table></td>
</tr></table></body></html>
```

3. 页面程序运行结果

用户发布留言的页面如图 10.18 所示。

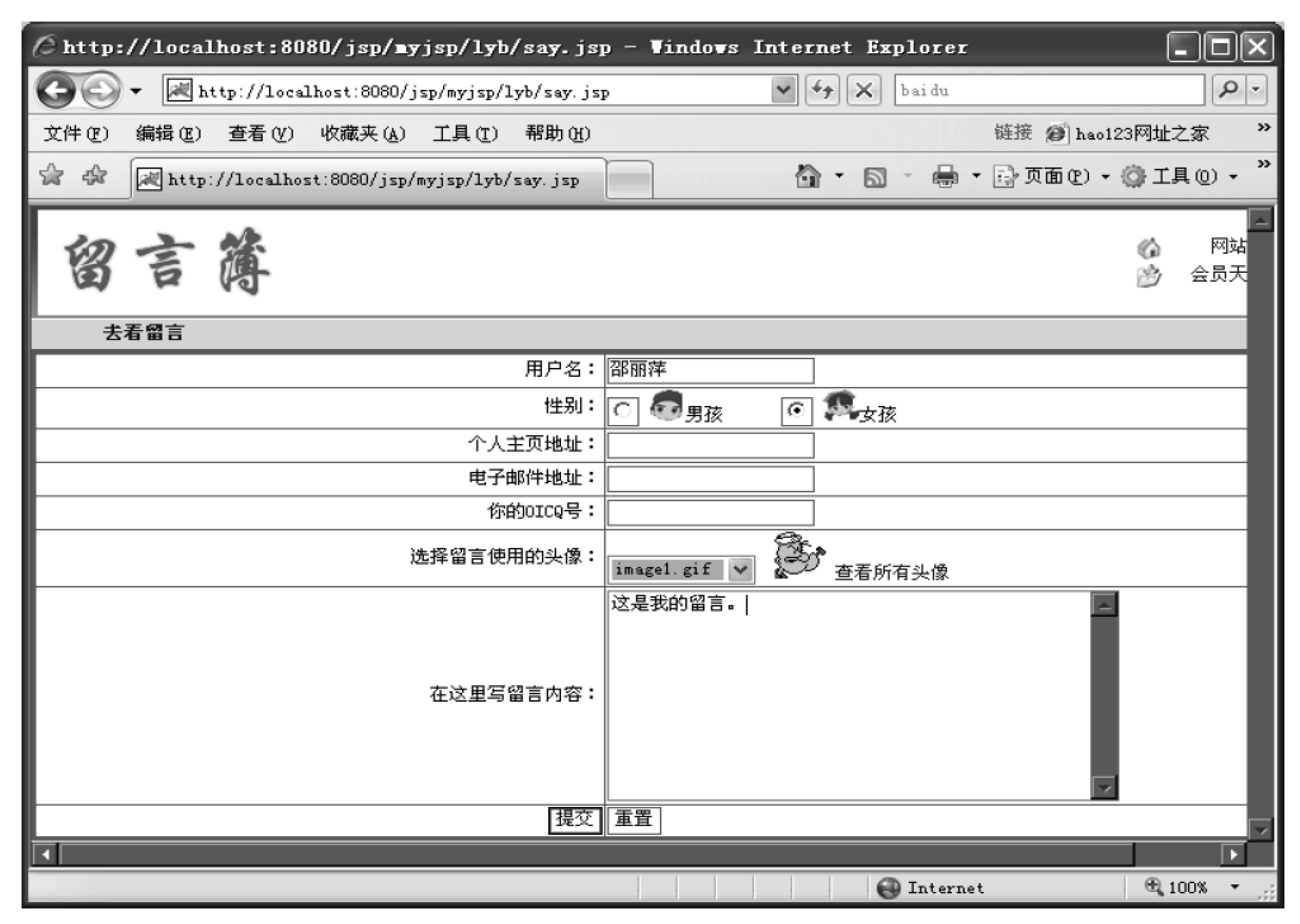


图 10.18　用户发布留言页面

10.6.8 保存用户留言与显示留言成功页面

1. saysucc.jsp 页面程序功能

在发布用户留言页面输入完毕单击“提交”按钮后，用户输入的数据将提交给保存数据程序 saysucc.jsp，该程序的功能是将用户输入的留言数据保存到数据库的 lyb 表中，然后显示留言成功信息，并提供了一个查看留言（返回留言主页）的接口。

2. saysucc.jsp.jsp 页面程序代码

保存用户留言与显示留言成功页面 saysucc.jsp 的源代码如下所示，可参考使用。

```
<%@page language="java" contentType="text/html;charset=gb2312" errorPage=""%>
<%@page import="java.sql.*"%>
<%@include file="opendata.jsp"%>

<html>
<head>
<link href="lyb.css" rel="stylesheet" type="text/css">
<jsp:useBean id="bean" scope="page" class="lyb.bean" />
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<script src="week.js" language="JavaScript" type="text/javascript"></script>
<img src="images/logo6gif.gif" width="160" height="35" border="0">
<img src="images/homepage.gif" width=16 height=16>
<a href="../main.jsp">网站主页</a>
<img height=16 src="images/Favorites.gif" width=16>
<a href="../user.jsp">会员天地</a>
<p>

<%
String IP=request.getRemoteAddr();
String username1=request.getParameter("name");
String username=bean.ex_chinese(username1);
String www=request.getParameter("www");
String oicq=request.getParameter("oicq");
String email=request.getParameter("email");
String memo=new String(request.getParameter("memo").getBytes("8859_1"));
String sex=request.getParameter("sex");
String face=request.getParameter("face");
username=username.trim();
www=www.trim();
oicq=oicq.trim();
email=email.trim();
```

```
String dd=bean.gettime();

sql="insert into lyb(name,IP,email,oicq,www,addTime,memo,sex,face) values ('"+username
+"','"+IP+"','"+email+"','"+oicq+"','"+www+"','"+dd+"','"+memo+"','"+sex+"','"+
face+"')";
s.executeUpdate(sql) ;
try {
  s.close();
  c.close();
}
catch(Exception ex) { }
%>
<br>
<%=username%>你好!你写的留言已经成功录入,<a href="lyb-index.jsp">
<font color="#FF0000">可以返回用户留言主页</font></a>查看你的留言。</p>
</body>
</html>
```

3. 留言成功后显示的页面

留言成功后显示的 saysucc.jsp 页面如图 10.19 所示。



图 10.19　保存用户留言信息及显示留言成功页面

4. 通过数据库保存图片存放路径

本程序提供了将简体中文数据保存到数据库的方式、以及通过数据库存放图片路径以便显示图片的方式(当然,要有相应存放图片的文件夹及图片文件)。通过这种方式可以展示多种产品的图片。

5. 到主页的接口

在保存用户留言与显示留言成功页面 saysucc.jsp 上单击“可以返回用户留言主页”超链接,将返回主页,在这里可以看到刚刚输入的留言内容,如图 10.20 所示。



图 10.20 新的留言页面

10.7 思考与练习

10.7.1 思考题

10-1 常用的公用文件有哪些？

10-2 首页有什么特点？

10-3 主页有什么特点？

10-4 如何确定其他页面文件在主页中的显示位置？

10.7.2 上机练习

10-1 创建一个简单的进入主页的首页。

10-2 实现书中介绍的导航条程序 dht.htm，与主页上部程序 main-1.htm 重新组合为一个能通向所有功能页面的主页上部程序 main1.htm。

10-3 编写一个主页底部程序 main2.htm。

10-4 修改 register.jsp 程序，在会员注册时要求必须填写用户名、密码、电子邮件地址等数据。如果这些数值为空，不能进行注册，给出不能注册的原因，让用户重新注册或不注册。

10-5 如果用户忘记了密码该怎么办？在用户登录入口添加一个“忘记密码了”的链接，编写一个帮助用户找到密码的页面。

10-6 根据书中介绍，编写两个漂亮的计数器，一个统计浏览网页的次数，一个记录访问网站的人数。提高一步编写一个显示注册会员人数的计数器。

10-7 创建一个模块化的主页。

参 考 文 献

[1] 邵丽萍,邵光亚.Java 语言程序设计(第 3 版).北京:清华大学出版社,2008.

[2] 邵丽萍,张后扬 等. 网页编程技术(第 2 版). 北京:清华大学出版社,2007.

[3] Paul Trebblett 著. JSP 即时应用. 高波等译. 北京:人民邮电出版社 2001.

[4] 邵丽萍,张后扬 等. 网站编程技术实用教程. 北京:清华大学出版社,2005.

[5] 邵丽萍. 网站开发实践实验教程. 北京:电子工业出版社,2007.

高等院校计算机应用技术规划教材书目

基础教材系列

计算机基础知识与基本操作（第 3 版）
实用文书写作（第 2 版）
最新常用软件的使用——Office 2000
计算机办公软件实用教程——Office XP 中文版
计算机英语

应用型教材系列

QBASIC 语言程序设计
QBASIC 语言程序设计题解与上机指导
C 语言程序设计（第 2 版）
C 语言程序设计（第 2 版）学习辅导
C++程序设计
C++程序设计例题解析与项目实践
Visual Basic 程序设计（第 2 版）
Visual Basic 程序设计学习辅导（第 2 版）
Visual Basic 程序设计例题汇编
Java 语言程序设计（第 3 版）
Java 语言程序设计题解与上机指导（第 2 版）
数据库应用技术（FoxPro）
Visual FoxPro 使用与开发技术（第 2 版）
Visual FoxPro 实验指导与习题集
Access 数据库技术与应用
Internet 应用教程（第 3 版）
计算机网络技术与应用
网络互连设备实用技术教程
网络管理基础（第 2 版）
电子商务概论（第 2 版）
电子商务实验
商务网站规划设计与管理
网络营销
电子商务应用基础与实训
网页编程技术（第 2 版）
网页制作技术
实用数据结构
多媒体技术及应用
计算机辅助设计与应用
3ds max 动画制作技术（第 2 版）
计算机安全技术
计算机组成原理
计算机组成原理例题分析与习题解答
计算机组成原理实验指导
微机原理与接口技术
MCS—51 单片机应用教程
应用软件开发技术
Web 数据库设计与开发
平面广告设计（第 2 版）
现代广告创意设计
网页设计与制作
图形图像制作技术

实训教材系列

常用办公软件综合实训教程
C 程序设计实训教程
Visual Basic 程序设计实训教程
Access 数据库技术实训教程
SQL Server 2000 数据库实训教程
Windows 2000 网络系统实训教程
网页设计实训教程（第 2 版）
小型网站建设实训教程
微型计算机及小型网络系统的安装与维护实训教程
网络技术实训教程
Web 应用系统设计与开发实训教程
图形图像制作实训教程

实用技术教材系列

Internet 技术与应用（第 2 版）
C 语言程序设计实用教程
C++程序设计实用教程

Visual Basic 程序设计实用教程
Visual Basic.NET 程序设计实用教程
Java 语言实用教程
应用软件开发技术实用教程
数据结构实用教程
Access 数据库技术实用教程
网站编程技术实用教程
网络管理基础实用教程

Internet 应用技术实用教程
Flash MX 动画制作实用教程
Dreamweaver 网页设计实用教程
多媒体应用技术实用教程
软件课程群组建设——毕业设计实例教程
软件工程实用教程
三维图形制作实用教程

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室　计算机与信息分社营销室 收

邮编：100084　　　　　　电子邮件：jsjjc@tup.tsinghua.edu.cn

电话：010-62770175-4608/4409　　　　邮购电话：010-62786544

教材名称：网站编程技术实用教程（第 2 版）

ISBN：978-7-302-19720-1

个人资料

姓名：________________ 年龄：_______ 所在院校/专业：_____________________

文化程度：____________ 通信地址：_______________________________________

联系电话：____________ 电子信箱：_______________________________________

您使用本书是作为：□指定教材 □选用教材 □辅导教材 □自学教材

您对本书封面设计的满意度：

□很满意 □满意 □一般 □不满意　改进建议______________________________

您对本书印刷质量的满意度：

□很满意 □满意 □一般 □不满意　改进建议______________________________

您对本书的总体满意度：

从语言质量角度看 □很满意 □满意 □一般 □不满意

从科技含量角度看 □很满意 □满意 □一般 □不满意

本书最令您满意的是：

□指导明确 □内容充实 □讲解详尽 □实例丰富

您认为本书在哪些地方应进行修改？（可附页）

____

____

您希望本书在哪些方面进行改进？（可附页）

____

____

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案（素材），有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案（素材），希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjjc@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（http://www.tup.com.cn 或 http://www.tup.tsinghua.edu.cn）上查询。